```
Ø136Ø PRINT TAB(15*(J-1)+1); N$(I);
Ø137Ø NEXT J
Ø138Ø PRINT
Ø139Ø NEXT I
Ø14ØØ PRINT
Ø141Ø LET K = K-1Ø
Ø142Ø RETURN
Ø143Ø END
```

```
THIS IS A MEMORY TEST!!
IMAGINE YOU HAVE A PACK OF CARDS. I SHALL ASK YOU TO GIVE ME A CARD EITHER 1) OF A
GIVEN SUIT OR 2) OF A GIVEN VALUE.
IF YOU GIVE ME A NEW CARD YOU SCORE MORE POINTS, BUT IF YOU TRY TO GIVE ME A CARD YOU
HAVE ALREADY USED I SHALL TAKE POINTS OFF.
IF YOU WANT TO SEE THE CARDS YOU HAVE USED TYPE 'STATUS' IN REPLY TO ANY QUESTION,
BUT YOU LOSE 1Ø POINTS FROM YOUR SCORE.
O.K. LET'S START.
PLEASE TYPE THE SUIT OF AN EIGHT                                    ? HEART
OK. SCORE =  1      PLEASE TYPE THE SUIT OF A THREE                 ? CLUB
OK. SCORE =  2      PLEASE TYPE A HEART                             ? ACE
OK. SCORE =  4      PLEASE TYPE THE SUIT OF A SIX                   ? CLUB
OK. SCORE =  5      PLEASE TYPE A CLUB                              ? TWO
OK. SCORE =  8      PLEASE TYPE A SPADE                             ? JACK
OK. SCORE =  9      PLEASE TYPE A HEART                             ? JACK
OK. SCORE = 12      PLEASE TYPE THE SUIT OF A SEVEN                 ? HEART
OK. SCORE = 13      PLEASE TYPE A HEART                             ? THREE
OK. SCORE = 18      PLEASE TYPE A DIAMOND                           ? TEN
OK. SCORE = 19      PLEASE TYPE THE SUIT OF A SEVEN                 ? SPADE
OK. SCORE = 21      PLEASE TYPE A HEART                             ? QUEEN
OK. SCORE = 27      PLEASE TYPE THE SUIT OF A SEVEN                 ? CLUB
OK. SCORE = 3Ø      PLEASE TYPE A HEART                             ? NINE
OK. SCORE = 37      PLEASE TYPE THE SUIT OF A FOUR                  ? DIAMOND
OK. SCORE = 38      PLEASE TYPE THE SUIT OF A TWO                   ? DIAMOND
OK. SCORE = 4Ø      PLEASE TYPE THE SUIT OF A KING                  ? CLUB
OK. SCORE = 41      PLEASE TYPE A SPADE                             ? NINE
OK. SCORE = 44      PLEASE TYPE A DIAMOND                           ? FIVE
OK. SCORE = 48      PLEASE TYPE THE SUIT OF AN ACE                  ? DIAMOND
OK. SCORE = 5Ø      PLEASE TYPE THE SUIT OF A QUEEN                 ? SPADE
OK. SCORE = 52      PLEASE TYPE A DIAMOND                           ? FOUR
THINK AGAIN!!       PLEASE TYPE A DIAMOND                           ? STATUS

CLUBS               DIAMONDS            HEARTS              SPADES
                    ACE                 ACE
TWO                 TWO
THREE                                   THREE
                    FOUR
                    FIVE
SIX
SEVEN                                   SEVEN               SEVEN
                                        EIGHT
                                        NINE                NINE
                    TEN
                                        JACK                JACK
                                        QUEEN               QUEEN
KING

PLEASE TYPE A DIAMOND                                               ? THREE
OK. SCORE = 33      PLEASE TYPE A DIAMOND                           ? SIX
OK. SCORE = 4Ø      PLEASE TYPE THE SUIT OF A KING                  ? DIAMOND
OK. SCORE = 42      PLEASE TYPE THE SUIT OF A QUEEN                 ? DIAMOND
OK. SCORE = 45      PLEASE TYPE A CLUB                              ? ACE
OK. SCORE = 51      PLEASE TYPE A DIAMOND                           ? SEVEN
OK. SCORE = 61      PLEASE TYPE THE SUIT OF AN EIGHT                ? SPADE
OK. SCORE = 63      PLEASE TYPE THE SUIT OF A THREE                 ? SPADE
```

CT

computing today

MAR 1979

50p

TANGERINE DREAM

UPMARKET VDU

S100 PRINTER

TRITON EXPANSION

STOMPER-A NEW GAME

MICROBIOLOGY-THE 6800

computing today

VOL. 1 NO. 1
MARCH '79

EDITOR: Gary Evans.
EDITORIAL DIRECTOR: Halvor W Moorshead.
EDITORIAL ASSISTANTS: Rick Maybury, Ian Graham, Henry Budgett.
ART DIRECTOR: Diego Rincon.
PRODUCTION: Pete Howells, Paul Edwards, Tony Strakas.
PROJECT DEVELOPMENT: Ray Marston (Editor), Steve Ramsahadeo.
READER SERVICES: Alan Carlton (Man).
ADVERTISING: Chris Surgenor (Man), David Dinfield, Joy Cheshire.

CONTENTS

EDITORIAL

With this issue Computing Today goes from its embrionic state as a supplement in ETI to a fully fledged magazine in its own right. With all the advantages in space and resources this brings we shall continue the trend set in the early stages of the Computing Today supplements plus a lot more.

The contents of this first issue reflect the diverse nature of the computing field with items ranging from detailed hardware features to a tutorial series on the BASIC language.

Our two constructional projects are, we believe, the first such projects to appear in a computing magazine with all the details necessary to complete the design within the pages of the journal. The S100 printer will provide valuable hardcopy at a price far below many commercial alternatives while the TRITON expansion will extend the capabilities of the already versatile DIY computer published in our sister magazine ETI.

On the software side we continue with our tutorial series on BASIC and publish a couple of programs written in this language. One of these is purely for fun, the intriguing Stomper, game while the other emphasises the educational potential of the computer with a Geography Test program.

Between these major items is spread a host of other features. Microbiology, with a detailed look at the 6800 MPU and an item looking at what the gentlemen from the Far East are doing with MPUs.

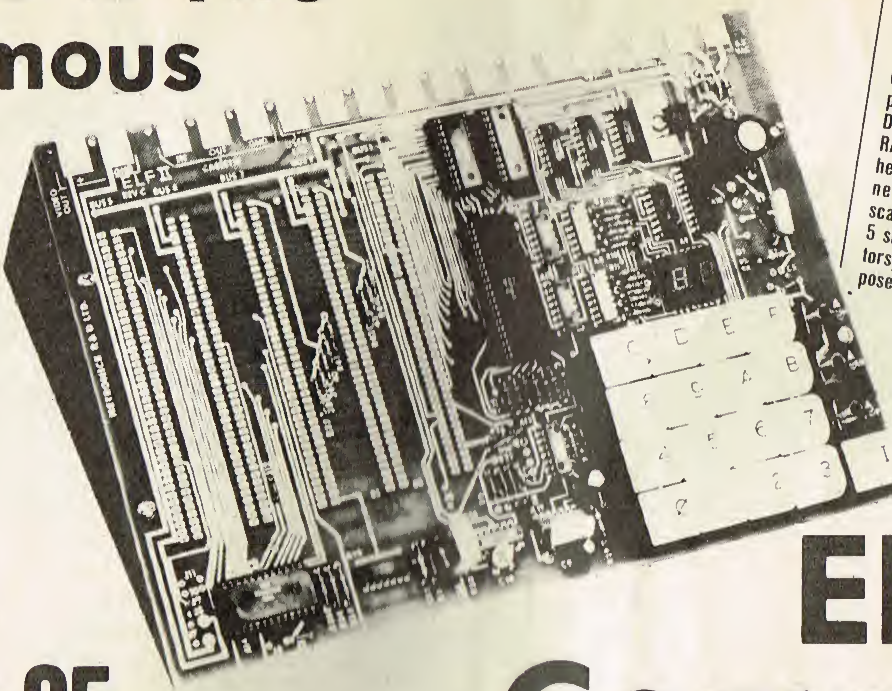
We hope you'll like the contents of this first issue but if we've got it wrong please write and tell us so. To those of you who have got systems we'd like to ask the question "what are you doing with them" — please write and tell us.

Gary Evans

Computing Today International is normally published on the third Friday of the month prior to the cover date.

EDITORIAL AND ADVERTISEMENT OFFICE
25-27 Oxford Street, London W1R 1RF. Telephone 01-434 1781/2 Telex 8811896

This Is The Famous



SPECIFICATIONS

The £99.95 ELF II computer features an RCA COSMAC COS/MOS 1802 8-bit microprocessor addressable to 64k bytes with DMA, interrupt, 16 registers, ALU, 256 byte RAM expandable to 64K bytes, professional hex keyboard fully decoded so there's no need to waste memory with keyboard scanning circuits, built-in power regulator, 5 slot plug-in expansion bus (less connectors), stable crystal clock for timing purposes and a double-sided, plated-through pc board plus RCA 1861 video IC to display any segment of memory on a video monitor or TV screen along with all the logic and support circuitry you need to learn every one of the RCA 1802's capabilities.

ELF 11 Computer

£99.95

PLUS 8% V.A.T.

Stop reading about computers and get your hands on one! with ELF 11 and our new Short Course by Tom Pittman, you can master computers in no time at all! ELF II demonstrates all 91 commands an RCA 1802 can execute and the Short Course quickly teaches you how to use each of the 1802's capabilities

ELF 11's video output lets you display an alphanumeric readout or graphics on any T V screen or video monitor and enjoy the latest T V games

But that's not all, once you've mastered computer fundamentals ELF 11 can give you POWER with add-ons that are among the most advanced found anywhere American IEEE chapters plus hundreds of universities and major corporations have chosen the ELF 11 to introduce their students and personnel to microprocessor computing!

Learn The Skill That May Soon Be Far
More Important Than Your College Degree!

The ability to use a computer may soon be more important to your earning power than college degree Without a knowledge of computers you are always at the mercy of others when it comes to solving highly complex business, engineering, industrial and scientific problems People who understand computers can command MONEY and to get in on the action, you must learn computers Otherwise you will be left behind

ELF 11 is the F-A-S-T Way to Learn
Computer Fundamentals!

Regardless of how minimal your computer background is now you can learn to programme a computer in almost no time at all That's because Netronics has developed a special Short Course on Microprocessor And Computer Programming in non-technical language that leads you through every one of the RCA COSMAC 1802's capabilities so you'll understand everything ELF 11 can do and how to get ELF 11 to do it!

All 91 commands that an 1802 can execute are explained to you step by step The text written for Netronics by Tom Pittman, is a tremendous advance over every other programming book in print

Keyed specifically to the ELF 11, it's loaded with "hands on" illustrations When you're finished ELF 11 and the 1802 will no longer hold any mysteries for you

In fact, not only will you be able to use a personal computer creatively you'll also be be able to understand computing articles in the technical press

If you work with large computers, ELF 11 and our Short Course will help you to understand what makes them tick

A DYNAMITE PACKAGE FOR JUST £99.95 plus 8% VAT!

With ELF 11, you learn to use machine language — the fundamental language of all computers Higher level languages such as FORTRAN and BASIC must be translated into machine language before a computer can understand them With ELF 11 you build a solid foundation in computers so you really know what you're doing no matter how complicated things get

Video output also makes ELF 11 unique among computers selling for such a low price Attached to your T V set, ELF 11 becomes a fabulous home entertainments centre It's capable of providing endless hours of fun for both adults and children of all ages! ELF 11 can create graphics alphanumeric displays and fantastic video games

Only a low cost R F Modulator is required to connect ELF 11 to your T V's aerial socket! (To order one see coupon below)

H.L. Audio, 138 Kingsland Road, London E2 8BY

ELF 11's 5-card expansion bus (connectors not included) allows you to expand ELF 11 as your need for power grows If you're an engineer or hobbyist you can also use ELF 11 as a counter, alarm, lock, thermostat, timer or for countless other applications

ELF 11 EXPLODES INTO A GIANT!

Thanks to ongoing work by RCA and Netronics, ELF 11 add-ons are among the most advanced anywhere Plug in the GIANT BOARD and you can record and play back programmes, edit and debug programmes, communicate with remote devices and make things happen in the outside world Add Kluge Board to get ELF 11 to solve special problems such as operating a more complex alarm system or controlling a printing press Add 4k RAM Board and you can write longer programmes, store more information and solve more sophisticated problems

Expanded, ELF 11 is perfect for engineering, business, industrial, scientific and personal finance and Tax applications No other small computer anywhere near ELF 11's low price is backed by such extensive research and development programmes

The ELF-Bug Monitor is an extremely recent breakthrough that lets you debug programmes with lightning speed because the key to debugging is to know what's inside the registers of the microprocessor and, instead of single stepping through your programme, the ELF-BUG monitor, utilising break points, lets you display the entire contents of the registers on your T V screen at any point in your programme You find out immediately what's going on and can make any necessary changes Programming is further simplified by displaying 24 Bytes of RAM with full address, blinking curser, and auto scrolling A must for serious programmers! Netronics will soon be introducing the ELF 11 colour graphics and music system — more breakthroughs that ELF 11 owners will be the first to enjoy!

NOW BASIC MAKES PROGRAMMING ELF 11 EVEN EASIER!

Like all computers, ELF 11 understands only "machine language" — the language computers use to talk to each other But, to make life easier for you we have developed an ELF 11 Tiny BASIC It talks to ELF 11 in machine language for you, so you can programme ELF 11 with simple words that can be typed out on a keyboard such as PRINT RUN and LOAD

"ASK NOT WHAT YOUR COMPUTER CAN DO . . . BUT WHAT IT CAN DO FOR *YOU*"

Don't be trapped into buying a dinosaur simply because you can afford it and it's big ELF 11 is more fun than "big name" computers that cost a lot more money

With ELF 11 you learn to write and run your own programmes You're never reduced to being a mere key punch operator working blindly with someone else's predeveloped software

No matter what your speciality is, owning a computer which you really know how to use is sure to make you a leader ELF 11 is the fastest way there is to get into computers Order from the coupon below!

H. L. AUDIO LTD., DEPT. ET1, 138 KINGSLAND ROAD, LONDON E2 8BY

NOW AVAILABLE FOR ELF 11

- [ ] Tom Pittman's Short Course on Microprocessor & Computer Programming teaches you just about everything there is to know about ELF 11 or any RCA 1802 computer Written in non-technical language, it's a learning breakthrough for engineers and laymen alike £5.00 post paid!
- [ ] Deluxe metal cabinet with plexiglasdust cover for ELF 11 £29.95' plus £1.50 p&p R F modulator for use with T V set £3.00' post paid
- [ ] GIANT BOARD kit with cassette I/O, RS 232-C/TTY I/O 8 bit P I O decoders for 14 separate I/O instructions and a system monitor/editor, £39.95 plus £1.00 p&p
- [ ] KLUGE (prototype) BOARD accepts up to 36 IC's £17.00 plus 50p p&p
- [ ] 4k static RAM kit Addressable to any 4k page to 64k £89.95 plus £1.50 p&p
- [ ] Gold plated 86 pin connectors (one required for each plug in board) £4.70 post paid
- [ ] Professional ASC11 Keyboard kit with 128 ASC11 upper lower case set 96 printable characters, onboard regulator, parity, logic selection and choice of 4 handshaking signals to mate with almost any computer £64.95' post paid
- [ ] Deluxe metal cabinet for ASC11 Keyboard £19.95' plus £1.50 p&p
- [ ] ELF 11 Tiny BASIC on cassette tape Commands include SAVE LOAD + − × / ( ) 26 variables A-Z LET, IF/THEN, INPUT PRINT, GO TO, GO SUB, RETURN, END, REM, CLEAR, LIST, RUN, PLOT PEEK POKE Comes fully documented and includes alphanumeric generator required to display alphanumeric characters directly on your T V screen without additional hardware Also plays tick-tack-toe plus a drawing game that uses ELF 11's hex keyboard as a joystick 4k memory required £14.95' post paid
- [ ] Tom Pittman's Short course on Tiny BASIC for ELF 11 £5.00' post paid
  Expansion Power Supply (required when adding 4k RAM) £19.95
- [ ] ELF BUG TM Deluxe System Monitor on cassette tape Allows displaying the contents of all registers on your T V at any point in your programme Also displays 24 bytes of memory with full addresses, blinking cursor and auto scrolling A must for the serious programmer! £14.95' postpaid

COMING SOON A-D, D-A converter, Light Pen, Controller Board, Colour Graphics & Music System and more!

'Plus VAT at 8% on price and postage

CALL OR WRITE FOR WIRED PRICES

H L AUDIO LTD , DEPT E T I
138 KINGSLAND ROAD
LONDON E2 8BY
(TEL 01-739 1582)
Sole European Distributors for Netronics
R&D Ltd U S A

YES! I want to run programmes at home and have enclosed [ ] £109.56 including postage and V A T for RCA COSMAC ELF11 kit [ ] £5.94 including postage and V A T for power supply (required) [ ] £5.95 including postage and V A T for RCA 1802 Users Manual [ ] £5.95 including postage and V A T for Short Course On Microprocessor and Computer programming

[ ] I want mine wired and tested with power supply RCA 1802 Users Manual and Short Course included for just £164.10 including postage and V A T [ ] I am also enclosing payment (including postage and V A T) for the items checked at the left

Total enclosed

USE YOUR [ ] ACCESS [ ] BARCLAYCARD
Interbank
Account No

Signature Exp date

CREDIT CARD PHONE ORDERS ACCEPTED. 01-739 1582

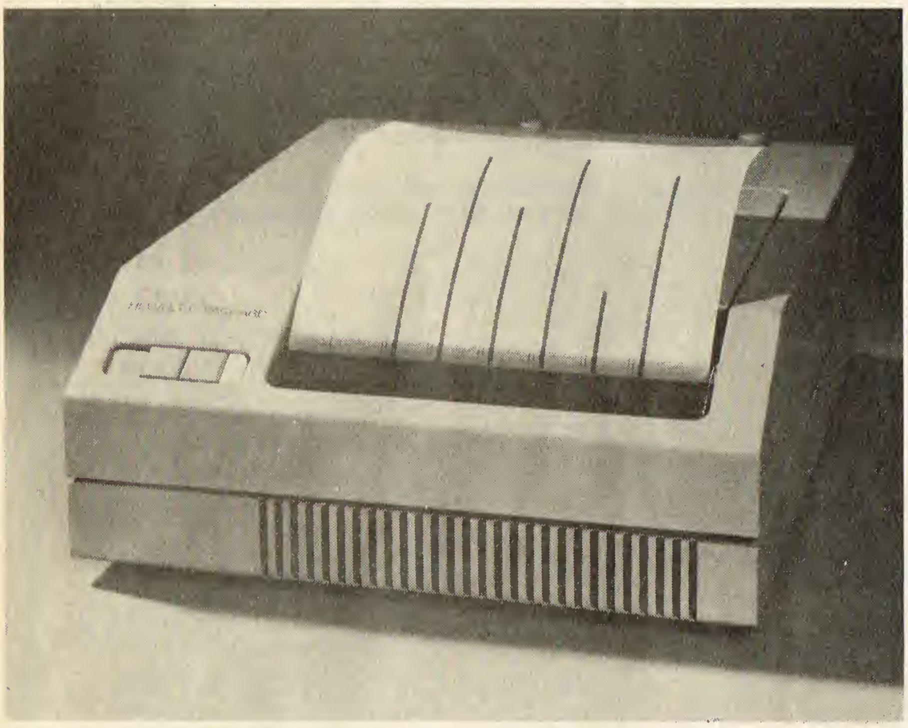
PRINT
Name

Address

DEALER ENQUIRIES INVITED

SEND TODAY

News



HOT PRINTER

An 80-character thermal graphics printer able to print at speeds up to 480 full lines per minute with high resolution, has been introduced by Hewlett-Packard. It is believed to be the fastest full-line thermal graphics printer presently available.

Intended for use in a quiet environment, the printer is simple to operate. Designated Model 9876A, it contains a standard ASCII 128-character set featuring both upper and lower case, and control characters. Each character is composed on a 5 × 7 matrix of 300-micron square dots. Two row positions above and below each character allow for special marks or ascending and descending characters.

Seven additional character sets, which exist in the printer at all times, can be accessed through software. These include French, German, Katakana, Spanish, Danish/Norwegian and Swedish/Finnish. Also, the user can create up to seven new characters at a time by defining special dot patterns which are then stored in the printer's memory.

SMART THINKING

The question of how we measure the intelligence of a machine is one that many people are giving a lot of thought to nowadays.

Dr. Chris Evans of the NPL has come up with six areas into which he feels a machine's ability should be broken down into and the overall intelligence computed from these. Rather like IQ tests measuring different areas of human performance.

We shan't go into much detail as to what Chris Evans means by each heading but perhaps it will set some of you thinking enough to write to us with your views.

Dr. Evans believes that a machine performance should be broken down into the following areas.

1. The machine's ability to capture data from its environment.
2. The machine's ability to store information.
3. The processing speed of the machine.
4. The speed at which the software of the machine may be changed.
5. The efficiency of the machine's software.
6. The range of the machine's software.

Any comments?

PET TO TELETYPEWRITER 33 BI-DIRECTIONAL INTERFACE

This interface package is a combination of hardware and software which provides two-way communication between a Commodore Pet and a Model 33 Teletypewriter. This not only provides the Pet with a printer, but, where the Teletypewriter is suitably equipped, with paper tape input and output and a full-size keyboard. Comprehensive documentation is included.

Hardware

The standard package is designed for a 20 milliamp current loop, but compatibility with 60 mA or RS232 can be provided by special order.

Connectors are provided to the Pet user port and the second cassette port but, by request, the package will be adapted to operate via the IEEE port. It is also possible to arrange for a second cassette unit to be operated simultaneously with the Teletypewriter.

A 'D' type connector to the Model 33 is fitted as standard.

Software

Two machine-coded programs are included in the package. One resides in the second cassette buffer, the other reserves RAM at the top of 8K and the coding resides there, so that the second cassette unit can be used. By special order programs can be provided for Pets fitted with extra memory.

After the interface program has been run, any programs may be loaded. These programs may send to and receive from the Teletypewriter by including short sub routines provided in the package. Detailed instructions are included.

A BASIC program is available as an extra giving a routine to list an entire program on the printer without operator intervention.

Allen Computers,
16 Hainton Avenue,
Grimsby,
South Humberside DN32 9AS.

NASCOM 1 AT MICRODIGITAL

The Microcomputer only shop providing a complete service from a single chip to a commercial data processing installation. Well worth a visit for a look around and a chat.

nm

NASCOM I............£178.20

From 10th February 1979.
Includes VAT and carriage.

The Nascom I was exceptional value for money at the old price, now it is unbeatable.
The Nascom I is the best possible introduction to the world of personal computing, yet it has the power and flexibility to be expanded into a full data processing system.
The specification includes powerful Z80 processor, parallel I/0 controller with two 8 bit ports. UART driving cassette interface or most serial peripherals, video output to plug in the ariel socket of your T.V., 2K bytes of RAM (IK user and IK video), proven IK byte monitor program in EPROM and a spare EPROM socket.
The kit is complete, all that is required is a power supply a domestic T.V. and a domestic cassette recorder.

POWER SUPPLIES
There are two power supplies available, a 3 amp supply which will power the basic kit and some expansion and an 8 amp supply with toroidal transformer which will power a very large system. Both supplies can be mounted in the vero frame.

3 amp P.S.U. kit.............................. £26.46
8 amp P.S.U. kit.............................. £64.80

EXPANSION
Nascom I is expanded by connection to a buffer board which creates a 77 way bus structure "NASBUS" into which expansion boards plug directly. The bus structure is carried along a motherboard which allows future boards to be added and to keep your computer neat the **Nascom I**, power supply, buffer board, mother board and expansion boards can all be mounted in a vero frame.

Buffer Board.......................... £35.10
Mother board.......................... £10.26
Mini Motherboard...................... £3.13
Vero frame.......................... £31.86

NASBUS
The 77 way **Nasbus** has the following advantages:—
1. Uses standard Veroboard as a motherboard and Standard 0.1" single sided edge connectors for expansion cards. These components are readily and cheaply available.
2. The bus structure leaves 8 spare data lines and 4 spare address lines for future use of 16 bit processors.
3. The power lines are regulated, on board regulators are therefore not needed which obviates the necessity for fan assisted cooling.
4. All cards use lower power, low noise shottky buffering which means the bus is quiet and does not need sophistications like active termination or interleaved ground planes.
5. Expansion boards are standard 8" x 8" vero DIP boards which are economic and give a good useable area.

All prices include VAT and Carriage

MEMORY
The memory expansion board can carry 16 dynamic RAM chips, these can be either 4K bit or 16K bit chips and the board is offered with8,16 or 32K bytes of RAM. The 16K board can be expanded to 32K by plugging in 8 more 4116 chips.
The memory expansion board also has room for 4 2708 UVEPROMS each of IK bytes and a lot of pre-programmed systems software is available to fit these sockets.

8K RAM board kit........................ £91.80
16K RAM board kit.................... £151.20
32K RAM board kit.................... £216.00
Set 8 x 4116.......................... £75.60
Additional 2708.......................... 11.34

INPUT/OUTPUT
For people wanting to use more peripherals than the standard kit allows for, **Nascom** are producing an I/O board which can carry a counter timer chip and a number of PIO's and UARTS. This will be available in March.

I/O board...................... £37.80
CTC.............................. £8.64
UART............................ £5.94
PIO.............................. £8.64

BASIC
To allow high level language programming **Nascom** have produced a 2K Tiny basic and a 3K Super Tiny Basic in 2 or 3 2708 EPROMS respectively. Also available is an 8K Microsoft precision floating point basic in 8 2708's which will be available in April on a single 64K bit ROM to fit the EPROM board.

Tiny Basic.......................... £27.00
Super Tiny Basic.......................... £37.80
8K Basic (8 x 2708).................... £108.00
8K Basic (ROM).......................... £43.20

EPROM BOARD
Available in March this board will carry 8 x 2708 UVEPROMS and the 64K bit ROM containing basic. The board can also be used for burning in 2708 UVEPROMS.

EPROM BOARD.......................... £43.20

GRAPHIC BOARD
***Allows high resolution graphics on your* Nascom I.** Contains 4K of RAM.

Graphics board........................ £102.60

MONITOR
Nascom have written a new monitor, T4 the most powerful yet available for this machine it contains many desirable features not found on any other monitor. T4 comes in 2 x 2708 to plug into the main **Nascom I** board.

Nasbug T4.............................. £27.00

FIRMWARE
A powerful editor assembler zeap is available to run under Nasbug in 3 x 2708 or on tape. ICL Dataskill have produced a letter Editor available in 2 x 2708.

Zeap (tape).......................... £32.40
Zeap (Eprom).......................... £48.60
Letter Editor.......................... £75.60

THE FUTURE
In the near future a mini-floppy disk system will be available with either single or double drive. These will probably offer in excess of ½ a megabyte and 1 megabyte respectively at prices that will allow even the hobbyist to have a large data base. To take full advantage of the business and scientific uses opened up by disks **Nascom** intend to release several high level languages. Looking further forwards **Nascom** is a developing product, and the fact that many thousands are now in use will ensure that the latest in computer technology will be available at a competitive price.

OPENING HOURS: ***9-5.30 Monday to Saturday. Friendly, expert staff always on hand!***

Phone in your Access/Barclaycard Number on
051-236-0707
or complete this order form

PLEASE SEND ME:
....
....
....
....
....

I ENCLOSE:
CHEQUE/POSTAL ORDER NO...............................
BARCLAYCARD NO...............................
ACCESS CARD NO...............................
NAME...............................
ADDRESS...............................
....
....
....

COMPLETE AND POST TO

MICRODIGITAL LTD. 25 BRUNSWICK STREET LIVERPOOL L2 0BJ Tel: 051-236 0707

News

CUSTOMS CURRENCY CONVERSION

Dear Sir,

Ref: Letter, page 24, Computing Today, February edition, from Customs and Excise.

I once ordered some goods from an American firm as a result of their advert in a British magazine, unfortunately this firm stated the value on the Customs declaration slip to be "20" with no units. As the package originated in America, British Customs decided that this meant $20 whereas the value stated on the invoice inside the packet was £20 as I had paid by Sterling cheque.

Now of course *I* am guilty of "making a false statement on a customs declaration" despite the fact that I made no statement at all and neither did the American company make any false statement, nevertheless Customs and Excise form Judge and Jury and I am clobbered for a "penalty" for my wrongdoing.

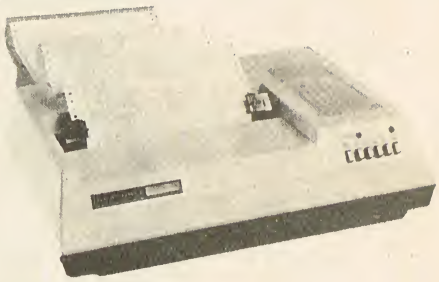
Would-be purchasers from America be warned, not only can you be "done" for VAT and duty but also for a penalty payment as well, purely due to some overworked customs official quoting the rule book.

Yours sincerely,
Peter G. Dunkley.

HEATH KIT KAT

The latest catalogue from Heathkit has a number of additions to their range of personal computer kits on display. Among them are floppy disks for both the H8 and H11A systems and what looks like the bargain of the year — the H14 line printer.

The printer has both upper and lower case capability and prints the standard 96 character ASCII set as a 5 × 7 dot matrix. Line length is selectable at 80, 96 or 132 characters while line spacing is six lines per inch with eight software selectable. Baud rates are between 110 and 9600. The printer allows the use of edge punched fan paper forms from 2·5in. to 9·5in. in width. In kit form this cornucopia of facilities will cost you just over £300. We've ordered one and will let you know in how many bits the kit is when it arrives.



Meanwhile if you want to know more about Heathkit's range of computers send them 20p for their special computer brochure.
Heath (Gloucester) Ltd.,
Gloucester GL2 6EE.

MORE BIG RAM

Motorola have announced a 65,536 × 1 dynamic RAM.

The next entry in the 64K dynamic RAM race, the Motorola MCM6664, has the impressive set of credentials that is typical of this new generation RAM.

—Single plus 5 volt power supply
—150 ns maximum access time
—250 mW maximum power dissipation
—16 pin 300-mil wide dual-in-line package
—fully TTL compatible

NEW FROM KALAMAZOO

The introduction of the K2000 small business computer marks the entry of Kalamazoo into the so-called "silicone chip" field. A fraction of the cost of a larger, main-frame computer, the K2000 brings advanced technology within the reach of smaller concerns.

With the K2000, however, Kalamazoo is offering more than just a computer. The company is selling a system: not just the hardware, but with it systems thoroughly researched in the market place, based on 80 years' business systems experience, mostly gained in the service of smaller companies. The K2000 has been designed to make systems work — not (as is often the case) the other way round.

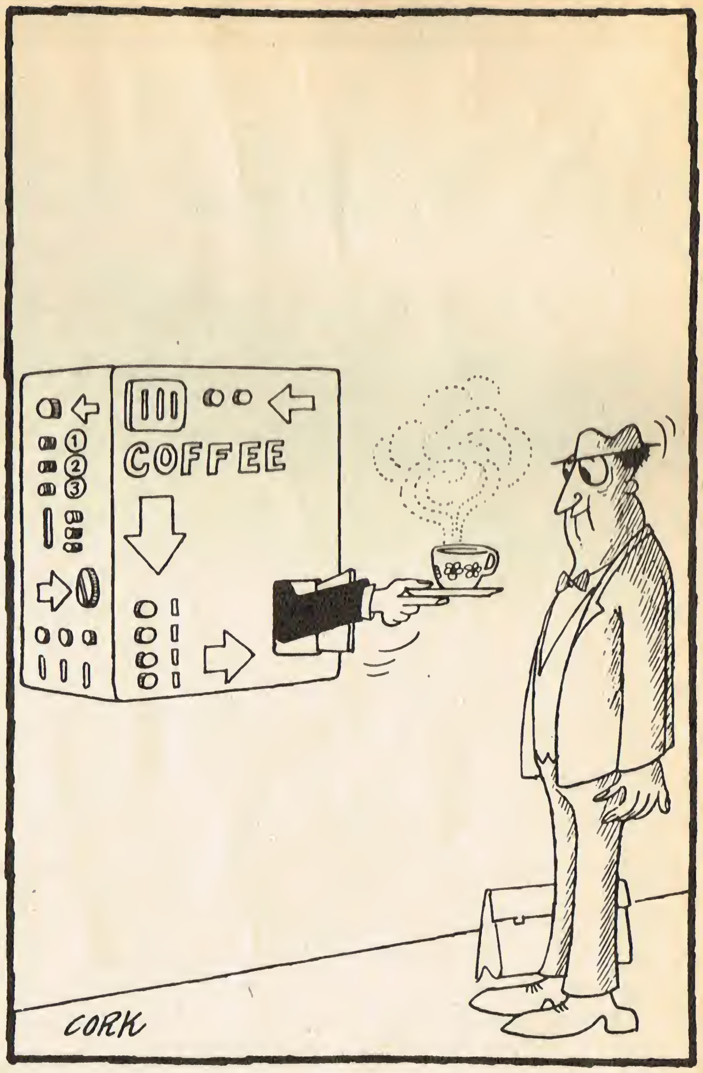
With the system Kalamazoo will provide a total service — hardware, maintenance of that hardware, software, including the tailor-making of systems to suit individual needs, the implementation of those systems, and training: not only for operators but also for middle and senior management personnel, to enable the latter to evaluate the system and obtain maximum benefit from the use of a computer.

GROWING UP WITH NASCOM

The NASCOM 1 DIY computer kit has been with us for some time now and was reviewed in the first of the Computing Today supplements printed in ETI. Until recently however there has not been any expansion for the basic NASCOM set up available. That has changed now. NASCO who manufacture the NASCOM 1 have introduced a buffer board plus expansion memory cards along with a Tiny BASIC in EPROM that many people have been eagerly awaiting. Meanwhile Comp. Components have introduced an S100 adaptor for the NASCOM 1 that will allow boards to this popular BUS structure to be hooked up to your NASCOM. Comp also have a low cost grpahics board available for the NASCOM.

We will be following up both these aspects of the NASCOM story next month but meanwhile for further details contact either
NASCOM MICROCOMPUTERS
121 High Street
Berkhamstead
Herts
or
Comp Computer Components
14 Station Road
New Barnet
Hertfordshire


CORK


COFFEE
CORK

electronics today international

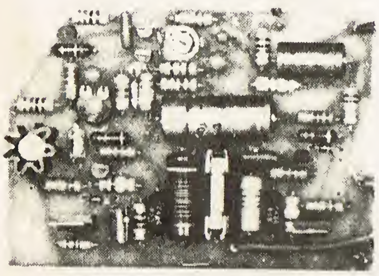
What to look for in the April issue: On sale March 2nd

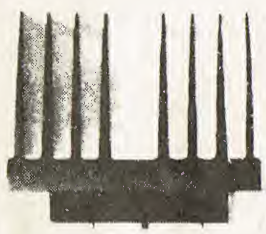
Amp Survey

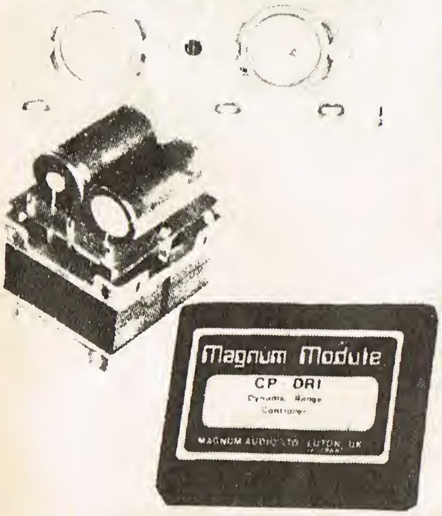
Build-it-yourself hi-fi continues to flourish, and new designs appear almost daily. Power amplifiers are a favourite in the field, and their numbers, by now, are legion.

Unfortunately there is no way for the home constructor to 'listen in' to a module before he builds it, and thus he is left to fall back on the spec. sheets. Fine if you like it, rotten if you don't.

Next month we're surveying the field, giving full details of all the models we can find, and putting the market leaders against top quality commercial equipment to find out how they sound.





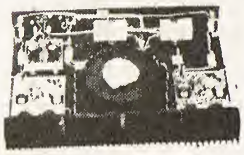


MAINS SEEKER

So you are about to drill the living room wall to hang up those shelves you promised the wife 7 years ago. Black & Decker in hand you advance to the plaster. Wait a minute there a mains socket right beneath.

Doubt sets in — to drill or not to drill — that is the question. Which way do the wires run? Will you black out the entire Universe if you try it? How can you find those wires?

Simple really — just read ETI next month when we have a neat little project to show you exactly where the mains wires lie!



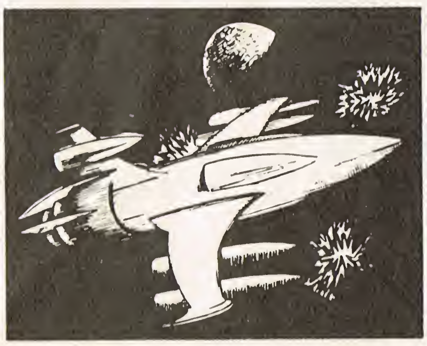
OCTAVE SHIFTER

A superb little circuit to add that instant 'jump' to guitar playing. Operated by a footswitch the effect has a unique sound all its own — not to be missed — no strings attached to this one.

3080

Well ten of them anyway The 3080 is a much under-rated device, and next month's IETs circuit man Tim Orr hopes to put that right with ten ways to use device, all comprehensively explained to help you design the other 3070 circuits yourself.

AMBUSH

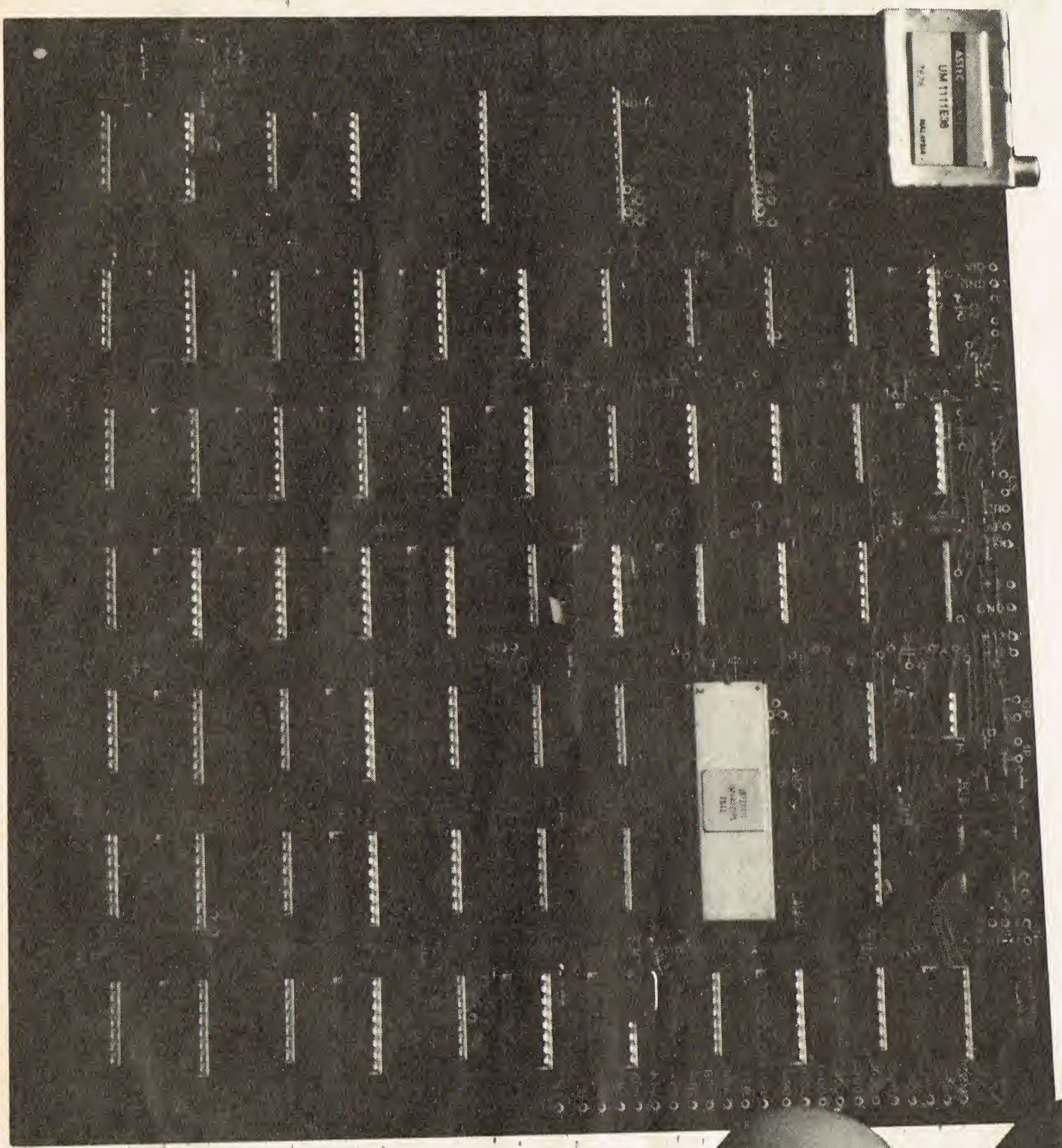


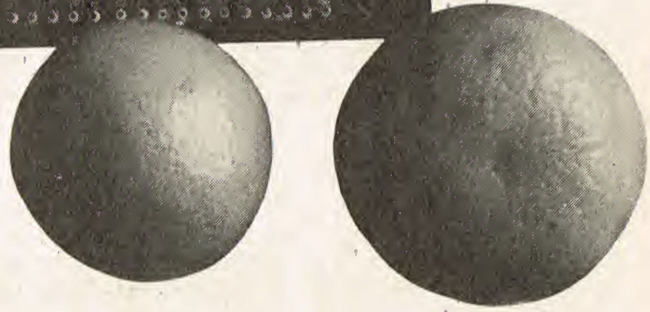
Your starship crashes through the void — running between the lines of enemy dreadnoughts to deliver medical supplies to the seiged plant of Tora. In order to preserve energy your ship has no weapons, only its shields and its speed.

Missiles can appear from any direction, and to destroy them you must actuate your shields at the precise moment of impact, thereby conserving power and allowing the engines to keep you moving at Warp Factor 20.

Can you make it through the Ambush and make Capt. Kirk look a cissy?

Tangerine Dream





With the current crop of single chip VDUs on the market the Tangerine 1648, with its mass of TTL devices, may seem out of place. This is not the case. The Tangerine offers a flexibility that is just not possible with the cheaper of the single chip VDUs, while the more flexible programmable devices will take some time to filter through to the amateur market and will be expensive at that.

TANGERINE SPECIFICATION

Before going on to describe the kit in detail a quick look at just what the Tangerine offers is in order. The display format is 16 lines of 48 characters. This format is rewirable to 16 × 44 in case of TV overscan, a common phenomenon that results in some VDUs losing the first and last characters of each line in the borders of the TV screen. Although a cure can usually be effected by adjustment of the TV's internal controls, most people are reluctant to go »»

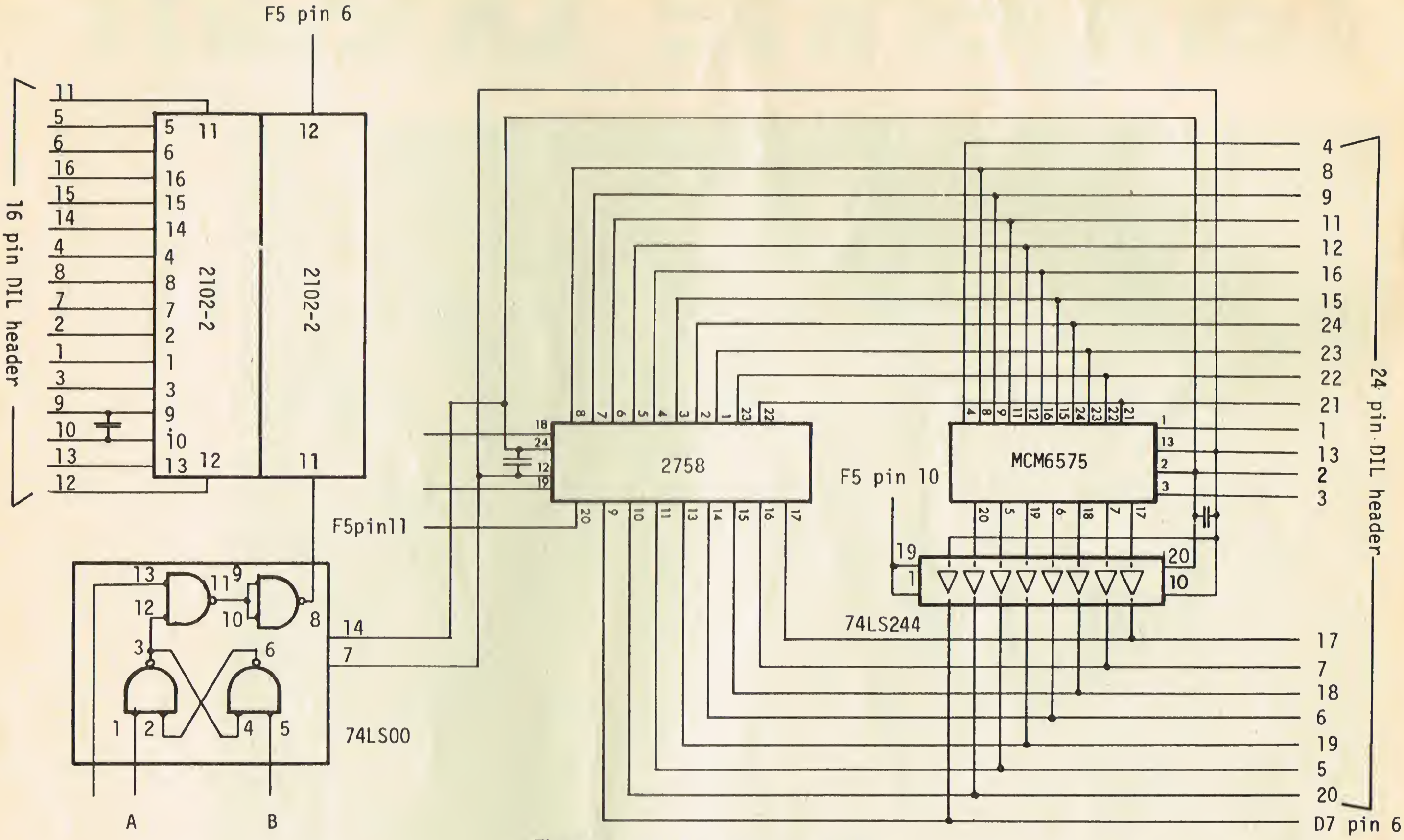


Figure 1 Add-On Graphics Circuit

delving into the insides of their TV making this variable line length a useful feature.

The character set comprises the 96 standard ACSII set displayed as a 9 × 7 dot matrix with downshift for certain lower case characters. The Tangerine's I/O interface can be configured as either RS232 or as a 20-60 mA current loop serial port. Baud rates are selectable between 75 and 19200, including of course 110 and 300.

Video output from the board is provided as a composite video output. There is also a modulated signal suitable for direct injection to the aerial socket of a standard domestic TV.

IN CONTROL

A number of control facilities are provided by the Tangerine. These are carriage return, line feed, bell, clear page, cursor up, left, right and home. Each of these control characters is wireable to any control character. The bell output is generated on board and consists of a 100 mS burst of approximately 1 kHz at 3 V p-p noload.

The Tangerine is made up from a number of circuit blocks. The UART and associated circuitry takes care of the parallel to serial and serial to parallel conversions. The on board RAM stores the character field as it is entered and is read out during a TV frame to build up the on screen display. The TV display is produced by a character generator ROM which in conjunction with control logic ensures that the character specified by the RAM is built up on screen. Another section of the VDU takes care of the cursor position. This position may either be altered in the normal way by a character input being detected (it is then incremented by one) or on command from the display control logic, which upon detecting a cursor control character will alter the cursor's position accordingly.

PACKAGE DEAL

Now to the kit itself. The Tangerine comes beautifully packaged, à la MEK D2, in a ring binder that carries all the documentation plus the PCB and components neatly sandwiched between its pages. This is not a kit for those who have never pointed a soldering iron in anger before, indeed the first thing pointed out in the construction section of the manual is that if the potential constructor is not confident in his ability to assemble the VDU he should return it for a refund, a ready built kit or find a friend to build it for him.

Having said that, anyone who can solder well should be able to build this kit with very few problems. The instructions are clear and unambiguous while the PCB is a high quality plated through product with a special mask that will prevent solder going where it shouldn't.

The documentation supplied with the Tangerine deserves a special mention. As remarked on above the standard of the text describing the construction of the unit is clear and precise. In addition to this section there is a detailed description of the complete circuit and how it works. Together with the complete circuit diagram and board overlay also provided this makes understanding what is happening inside the Tangerine possible. This knowledge is invaluable during fault finding or modification of the basic machine. Such a refreshing change from those scruffy, half complete notes and circuit diagrams that one so often comes across nowadays.

When construction is complete it is only necessary to

Tangerine Dream

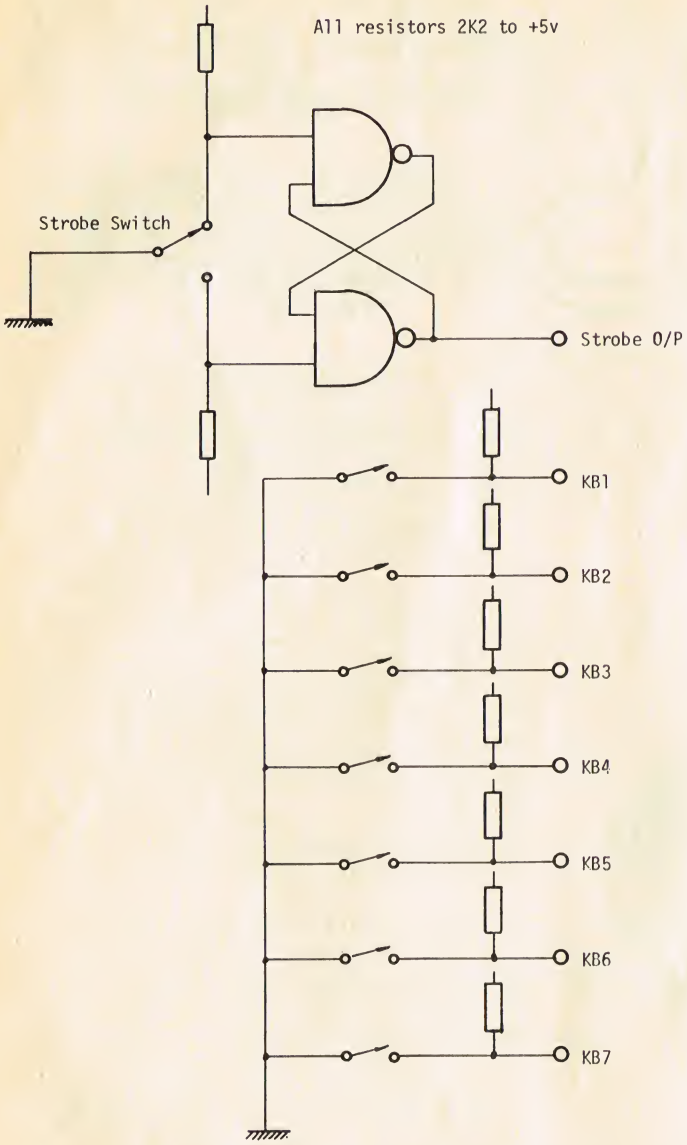


Figure 2 Keyboard Test Circuit

connect up a 5 V 800 mA, 12 V and −12 V 50 mA supply and the VDU is ready for operation. Connection details for the various options given above are provided and hooking the Tangerine up to your micro should present no problems.

In addition to the standard micro-VDU-TV set combination, the manual details a number of other applications. Driving seven segment LEDs and operating mains controlled devices are two of the options described.

Items in this section also include hints on how to make up for the fact that as the VDU is of the memory mapped type, there is no addressable cursor. Follow the software hints in this section and that can be overcome. Connecting the Tangerine in a parallel mode is also detailed along with an interesting section on adding a home made graphics ROM.

ROUNDING OFF

The Tangerine represents a high quality VDU system that provides a versatility in operation that is not found in many of the other VDUs on the market. The quality of components is high and when complete the Tangerine should go on to give reliable service for a long time. **CT**

The exciting new

TRITON

Personal Computer exclusively from:

TRANSAM

Complete kit of parts available only £286 (+ VAT)

Basic in Rom: a powerful 2k Tiny basic resident on board, makes Triton unique, easy to use and versatile.

Graphics: 64 Graphic characters as well as full alpha numerics.

Single Board: Holds up to 8k of memory, 4k RAM and 4k ROM, supplied with 3k ROM and 2k RAM.

Memory Mapping: 2 mode VDU, I/O or memory mapped for animated graphics.

Cassette Interface: crystal controlled Modem tape I/O with auto start/stop + "named" file search.

UHF TV Interface: On board uhf modulator, plugs into TV aerial socket.

Comes Complete with KEYBOARD, CASE, POWER SUPPLY, THRO-HOLE PLATED QUALITY P.C.B. FULL DOCUMENTATION POWERFUL 1k MONITOR & 2k TINY BASIC PLUS ALL COMPONENTS INCL. IC SOCKETS. NOTE TV SET & CASSETTE NOT INCLUDED.

EXPANSION: expand your triton simply and easily with our new 8-slot motherboard — complete with its own p.s.u. takes 8 plug in eurocards. Plug in 8k ram card now available.

EXPANSION MOTHERBOARD £50

8k STATIC RAM BOARD £97

Full details available in our brand new 1979 computer products catalogue — so order your copy now!

TRANSAM

I am interested in the Triton

Name ____________________

Address (please print) ____________________

____

____

CT1

BARCLAYCARD, ACCESS, VISA & MASTER CHARGE ARE WELCOME. SEND YOUR CARD NUMBER WITH ORDER

To TRANSAM COMPONENTS LTD.
12 CHAPEL STREET, NW1

Please send me the following

1. A copy of your latest catalogue I enclose 30p + S.A.E. 9 × 7 ☐
2. A copy of the Triton Manual I enclose £5 + 70p P & P. ☐
3. A complete kit of parts for Triton Computer, £286 + VAT + £4 P & P ☐
4. A Printed Circuit Board £50 + VAT & £1 P & P ☐

TOTAL ENCLOSED £ ....

Cheque, Money Order, etc.

All components can be bought separately and eleven packs can be purchased on an easy-to-buy scheme. See catalogue.

The P.C.B. alone is £50 + VAT plus £1 for packing and postage.

VAT rate is 8% on all kit components.

TRANSAM

TRANSAM COMPONENTS LTD.
12 CHAPEL STREET
LONDON, NW1 TEL: 402 8137

NEXT TO EDGWARE ROAD TUBE STATION MET LINE.
TRITON COMPUTER IS THE TRADE MARK OF TRANSAM COMPONENTS LTD.

Bits, Bytes and BAUDs

It seems that every radio amateur is talking about some kind of data transmission or other. There are two standard ways of sending data over radio used by amateurs today. Baudot, named after its inventor, was the earliest form of Teletype code and is still used internationally today, both in commercial and amateur service.

Baudot

Baudot comprises of five "bits" or levels of information per character, which are sent one bit at a time. Each bit is in either the "on" or "1" state or the "off" or "0" state, and in Teletype this represented by a "mark" or "space". "Mark" refers back to the days when morse was recorded on paper tape, and refers to the line being energized, thus marking the paper. Space indicates no current.

Over radio circuits, these marks and spaces are translated into audio tones, such as 2125 Hz and 2975 Hz. The difference between the frequencies of these tones is called the "shift".

If the bits from each character were just sent out right after the bits from the previous, very careful count would have to be kept at the receiving end to determine which was the last bit of one character and the first bit of the next. In practice this would lead to the impossible situation where one noise burst would lead to the destruction of the entire message!

To counteract this problem, "start" and "stop" bits were introduced. A start bit is always a one-bit transition from mark to space and back. This tells the receiving machine to receive the next five bits and decode them as data. When the data bits are sent, the sending machine restores the line to the mark condition, where it will stay until the next start pulse comes along. This means that when the machines are sitting idle, there is always current on the line. When you are sending at the maximum rate, there is always at least 1½ bits of STOP time (STOP time is when the line is MARKING after each character). This allows the receiving machine to get ready for the next character.

This START/STOP mode of operation is the simplest form of mechanical telegraphy signal, and most widely used at slow and medium speeds. Since any timing errors in the receiving machine are compensated for in the STOP time, there is no need to synchronise machines, other than their speed. For this reason, this method is called ASYNCHRONOUS transmission.

Asynchronous

Asynchronous transmission has few disadvantages — and they only become apparent at very high speeds. One such problem is the inefficiency of wasting time sending the START and STOP bits, when they are not needed for data purposes. However, at speeds used by amateurs, this is a small price to pay for the integrity of data.

One problem that became apparent with the proliferation of computers, and special codes for weather symbols, etc., was the limited number of codes able to be transmitted by Baudot. If you figure it out, there is a maximum of 2^5, or 32 codes possible with Baudot, so how do we code anything beyond that? The answer lies in the use of the LTRS (letters) and FIGS (figures) keys. These keys give the Baudot system 64 characters. Each key has two characters, lower and upper.

When the LTRS key is pressed, a character is sent which tells the receiving end that the codes that follow are ordinary letters. When it is desired to send numbers you have to precede them with the FIGS character. Each of these FIGS and LTRS signals is a full character and takes a whole seven-and-a-half bit time to send. As you can see, this could severely reduce the actual speed of transmission if you had a lot of letters interspersed with a lot of numbers or figures.

This became apparent over years of use, and led to the introduction of ASCII, the second code used by amateurs (American Standard Code for Information Interchange). This code can legally be used by Canadian amateurs but is restricted in many other parts of the world, including the U.S. It consists of eight bits, of which seven are actual data and one is a PARITY BIT, or check bit. This PARITY BIT is either sent as a MARK or SPACE, to make the number of MARK bits EVEN. By counting the number of MARK bits and checking to see if the number of them is EVEN, the receiving station has a pretty good idea if any data has been damaged by noise. Not all systems use the parity bit, in which case it is usually a MARK. The actual character coding only uses five bits still, and the sixth bit tells whether or not the code sent is a letter or a figure; its presence as a MARK indicates that this character is a *figure*. Thus FIGS and LTRS keys are not needed.

This leaves one bit to be discussed, and this is called the CONTROL bit. This bit must be a MARK for all PRINTING functions, i.e. all normal characters that will be printed at the other end. If you want to send a code into a computer, for instance, to tell it that the words to follow constitute the address of a message, you can send a character that could normally be a part of the message, but drop the CONTROL bit. The computer will not include the character in the message, but will understand that you want to tell it that the address of the message follows. This can be very handy as it means you can send control codes into a computer and not have to worry about the computer accidentally reading out part of the message as a control code. In the older Baudot system you had to make up weird combinations of three or four characters that could not normally be found in common English.

Some examples of common usage: »»

ASCII	BAUDOT	MEANING
control D	"NNN"	end of message
control P	"figs figs HH'	end of address

Distributor

So we have a Teletype machine sitting there waiting for some signals to turn it on. As you will know, all the action starts when the line goes open for a moment (9.09 milliseconds at 110 baud). This moment is called the *start time*, and it readies the machine to receive the character. From hereon I will refer to a speed of *110 baud* when I mention any timings, since this is a standard speed in data communications. Immediately after the *start bit* has finished, there follow eight bits of data. Each one of these data bits is strobed into the machine by a *rotating distributor* which was started by the start bit. The result is that the code bars will be set or reset in the machine at the correct time. For instance, when the signal condition representing the first bit is presented to the machine along the signal line, the distributor arm will be touching the connection to the circuitry for the first bit and it will be conditioned to either *mark* or *space* by the signal. At the end of this bit, the arm will have moved around to the beginning of the copper plate that is connected to the circuitry for the second bit, and so this circuitry will be conditioned to either the mark or space state by the signal from the line. And so, in this manner, the state of each bit along the line will be sent to a different part of the Teletype machine by the rotating distributor, and at the start of the stop bits, the machine will mechanically turn the data into a printable character.

The keyboard works in a similar manner, except in reverse. As soon as you press a key, in effect you are setting eight little switches to either the closed or open state. Moments later, the keyboard rotator starts rotating (what else would you expect it to do?) and sends the condition of each of these switches in turn as either a mark or space along the line, each bit taking the customary 9.09 milliseconds.

Definitions

What I have described above is the simplest form of serial-to-parallel conversion and parallel-to-serial conversion. In modern telecommunications equipment, these mechanical functions are replaced by solid-state logic. After telling you that, I think a few definitions are in order. *Parallel data* is data that is presented simultaneously on eight wires. These wires, for instance, could be connected to the eight switches on the keyboard that I mentioned earlier. It would be the simplest thing in the world to just connect these wires to the eight electromagnets that condition the mechanical bars of the printer, and in fact this is done in some computer sites where there are short distances involved. This is called *parallel transmission*. However, things being what they are in the business world, money comes first, and it would be eight times as expensive to string eight channels across the country as just one, so the serialization idea came into effect. *Serial data* just means that the bits are sent out one after another all on one wire as described above.

I mentioned earlier that this *serial, asynchronous* method is used most universally on low speed circuits. The reason is that mechanical equipment proliferates and this cannot be adjusted as finely as electronic equipment. The reason for the two start bits is to allow the mass of the rotor to come to rest and stay there awhile before going off on another trip around the circuit. Because the receiving machine starts each cycle at the same time as the sending machine, a slight variation in the speed of the receiving machine would not be serious.

Rate

Each bit in the above example takes 9.09 milliseconds to send, so it would seem only logical that to get the number of bits per second one would simply divide this into one second, and arrive at 110. However, it is not that simple. There is such a quantity, but it is called the *baud rate*. The actual name "bits per second" has been defined as the number of data bits that can be sent at this speed. As you will remember, for every eight bits just to keep the machine happy. These bits cannot be counted as data bits because they cannot be visibly seen to do anything at the other end. While the system is sending 110 bits per second, only eighty of them are data bits, so if you were to refer to this speed in bits per second, the value would be eighty BPS.

At slow speeds, this terminology is rarely used, since the *baud rate* is more meaningful in *asynchronous transmission*, because it relates more closely to the scientific quantities involved, whereas a businessman would be more interested in how soon he could get the latest Dow-Jones figures, so he would be more interested in the bit rate. To the uninitiated, it is just like comparing RMS power to music power, or peak power, by the hi-fi salesman.

As speeds get faster, mechanical monsters are replaced with solid-state equipment. Since there is no moving rotor to slow down and start again, many of these machines reduce the number of stop bits to one. The machine knows that the tenth bit after one start bit will be the start bit of the next character. In this case, the baud rate would be only slightly higher than the bits per second rate because 33 per cent of the dummy bits have been eliminated.

To take this one step further, you could completely eliminate the start and stop bits. When you do this, however, you are changing things just a little too far, and you do not have asynchronous transmission anymore. You now have *synchronous* transmission. This is only used at very high speeds because any small error, would require the resending of the whole block of data.

Modem

Now that you know what a teletype signal is, how it becomes a series of pulses, and how these pulses are timed, wouldn't it be nice if you could send them to somebody and have some device at the other end make them into teletype signals again?

Well, this is accomplished by a device called a Modem. The word MODEM is a contraction of MOdulator-DEModulator. The modulator portion takes the teletype signal from the teletype machine and converts it to two tones. When there is no current in the loop, a tone designated a "space" is sent. When current flows, a tone designated as "mark" is sent.

On the normal amateur teletype channels, such as on the short wave bands, these two tones are 2125 Hz and 170 or 850 Hz above it. (Both with respect to the carrier frequency, which is usually suppressed). At this point I would like to break from standard nomenclature. On UHF and VHF, we have the unique ability to communicate full-duplex (both ways at the same time). If both stations are using the same tones, difficulties will arise because under some circumstances a receiver may pick up signals from its own transmitter causing a garbled printout at the originating station. This problem arose many years ago in the North American TWX network (Teletype Writer Xchange) service. A standard was devised using two separate pairs of tones, one for use by the

ORIGINATING station, and one for the use ANSWERING station. When stations are listening for calls, they are in the ANSWER mode, listening on the pair of tones that the ORIGINATING station is sending on. (1270 Hz mark/1070 Hz space).

While the originating station is sending using 1270 Hz and 1070 Hz, it is also listening on the answer mode transmit frequencies of 2225 Hz for a mark and 2025 Hz for a space.

If station A originates a message to station B, and station B decides he is getting a wrong message, or the printout at his end is garbled, station B can talk back to station A without waiting for him to end his message.

Remember a while ago I talked about the parity bit? If a computer is sending some data to another computer, this full-duplex arrangement will allow the receiving computer to tell the sending computer about any parity errors as soon as they occur. On receipt of this interrupt, the sending computer needs only to re-send the bits that the listener did not get correctly, without having to wait until the end and re-send the whole block.

Another use of this full duplex operation is the so-called ECHO feature used by most computers. When you send a character to a computer, the computer "echoes back" the character. The character prints on your printer only after it has been to the computer and back, via the full-duplex modems. You can thus immediately tell what the computer received — a feature very handy if you are loading programs and want to be sure that the remote computer got your typing correctly.

We have explained the various forms of mechanical telegraphy that have evolved and how signals from these devices can be sent over very long distances by wire and radio. Until the era of the microcomputer, these signalling systems were limited in their use by the amateur to the sending of messages (and sometimes very cleverly-shaded pictures, using light and dark letter and figure combinations) between two points. Despite the limitations of the Baudot code, and the relative inflexibility of discrete logic, amateurs the world over developed systems for automatically turning on their machines, if and when they were specifically called. Thus, the "autostart" net was formed, a relatively little-known net that generally meets 75kHz up from the bottom edges of some amateur bands, notably 20 and 80 metres.

However, these logic circuits can be very frustrating and time-consuming to design — a fact which has been greatly helped by the availability of the microprocessor to amateurs.

Going back to the previous example of RTTY, an autostart unit consists of many TTL chips, some for decoding the incoming RTTY, some for timing, others for comparing the characters received with the station's callsign etc. Once the unit is built, it represents an investment of one to two-hundred dollars in a little black box that can only act as an autostart unit. Suppose the amateur then wants to add a Baudot-to-ASCII converter to his setup — he must invest another $100 in a unit that will perform that service for him, and so on until he has a station full of black boxes that are, all together, probably worth more than his main r.f. transceiver.

Now, with a microcomputer he can replace all those gadgets, save himself a lot of money, and at the same time have a very flexible station that can be easily reprogrammed at any time to operate in a slightly or a »»

completely different manner. The whole idea of the microcomputer is that the actual wired hardware is shared between the many functions of the station and it is the easily-changed program that connects the system up in the manner desired for a particular operation.

What is a microcomputer?

Basically, a microcomputer is a device for taking bits of information out of a "memory", performing a series of operations on these bits of information (called "data") as directed by the program, and putting the data back into memory, either in the same place or a different place, as the circumstances require. Sometimes the memory that it is working on is actually a set of flip-flops attached to some external device, sometimes the memory is "core" memory, sometimes it is one of its own internal registers. Either way, there is usually an input, some arithmetic or logic operations, and an output. Sometimes the data are written back into the same memory location as they were read from, thus destroying the original contents of that location, and sometimes they are written into a different location, so that the original data are preserved.

In each case described above, what is happening is a *transfer* of data. It can be said that *every instruction that a computer performs is a transfer of one kind or another.* The data may be modified *logically* (see Fig. 1) or *arithmetically* (see Fig. 2), but they are always *transferred.*

What's in a microcomputer?

As we can see from above, *memory* is always handy to have around. Without it, the microcomputer cannot store a program or data.

We also see that we must have some way of getting information from memory into the computer, and getting the results out of the computer and storing them in memory. This is called I/O (Input/Output).

In order to perform arithmetic and logical changes to the data, while it is flowing from memory, through the processor and out again to the memory, we need an *Arithmetic-Logic Unit* (ALU).

Finally, we need a master controller to put the whole act together and keep everything running smoothly. This is the *control* section.

In most microprocessor units (MPUs) available today, all of the above functions except memory have been included on the one chip (e.g. M6800, 8080, Z80, etc.) — see last month's Microbiography article for more information. All that is required is to add memory, put a program in it, and you have a microcomputer. (Actually, some chips have memory built right onto them, so that the whole system can be made very cheaply, but this is presently restricted to units that are made in the thousands because of the extremely high cost of mask-programming the computers when they are made.)

Fig. 1. 'AND'ing two memory locations together.

Computer compares each bit in location A to each bit in location B — if they are both in '1' then result is 1

LOCATION A		LOCATION B		RESULTING PRODUCT R
1		0		0
0		1		0
1		1		1
1	∩	1	=	1
0		1		0
0		1		0
1		0		0
0		1		0

Mathematical Notation $A \cap B = R$

Fig. 2. Arithmetic adding of two memory locations.

Computer adds each bit in location A to the corresponding bit in location B. If they are both 1, a carry is generated to the next bit down.

LOCATION	A	0 0 0 1 1 0 0 1
LOCATION	B	+0 1 0 1 1 0 0 0
LOCATION	B	=0 1 1 1 0 0 0 1

Mathematical Notation $a + B = C$

Bytes, words and data path width

The commonly-used microprocessors are generally "8-bit" machines. This means that every memory location, or *address*, can be visualised as being eight separate flip-flops, each of which can be set to the "1" state (ON) or the "0" state (OFF). Also, every circuit in the I/O and ALU sections is duplicated eight times, so each bit coming out of memory or going back into it is operated on by its own, dedicated circuits. Thus the operations happen concurrently, or in *parallel. Data path width* is the number of bits that can be acted upon in this parallel manner by a microcomputer.

Just as you may have already guessed, there also exist serial machines which do not have the multiple circuitry, and must handle data a bit at a time. These machines do have a place in data processing and are not to be scoffed at because of their apparent lack of sophistication, because in many applications they are fast enough for the job and they save the cost of the extra circuitry. (A few dollars a unit can really add up if you make several thousand units.)

Where the numbers fit in

This is all very well, but a computer's not much good unless you can put numbers in and get numbers out. So far, all we've got is a bunch of flip-flops that can be set and cleared in a zillion different ways like Christmas tree lights. The key to the whole matter is in the so-called *binary numbering system.*

When we were children, we learned how to count up to ten. When we reached ten, we had to remember that we had been through our hand once, and start again at 1 — but this time it was 1 for the one in our head and 1 for the finger, making eleven. When we grew older, we realised that the world doesn't end when we have ten fingers up and ten imaginary fingers stored in our head, and thus we mastered the magical transition from ninety-nine to a hundred, this time storing the big 1 in our toes.

This is fine for us humans, but the poor little computer only has one finger. However, to make up for this handicap, he is blessed with eight hands, each with one finger.

The finger (bit) can either be up (1) or down (0). Let's see how he counts.

Finger number Human Count:	7 6 5 4 3 2 1 0	(We always start at zero when dealing with computers)
	U = Finger up D = Finger down	
0	D D D D D D D D	
1	D D D D D D D U	
2	D D D D D D U D	
3	D D D D D D U U	
4	D D D D D U D D	
5	D D D D D U D U	
6	D D D D D U U D	
255	U U U U U U U U	

Every time the maximum number is reached in a column, we carry one into the next column on the next count. Isn't this just the same as carrying into the tens column when we get to nine, or the hundreds when we get to ninety-nine?

As you can see, our clever little computer can count up to 255 in human terms. (This is actually a count of 256, since we started from zero.)

So a computer with an eight-bit data path can store a number up to 255. What about a computer that has 12 or 16 bits? Well, we see that in the case of 8 bits, the value 2^8 is 256 — our old friend!

If we look at 2^{12}, we see that a twelve-bit machine can store a number as big as 4096. This is still not very impressive, and we must go to a 16-bit machine before we get a nice big figure, 65,536.

Let's assume we don't want to spend the extra money for a 16-bit machine to give us the 65,536 (usually referred to as 65K) storage capability, but we might occasionally want to do arithmetic using sums this large. The answer is to break up the numbers into two chunks (bytes) of 8 bits each, called the lower byte and the upper byte, and make up a 16-bit word. This is all very well, but we still need to add the numbers 8 bits at a time. This is usually taken care of by a bit called the *carry*. The carry bit is made a 1 if the addition of the two 8-bit numbers in the lower byte makes a number that is too large to store in an 8-bit memory location. What happens in this case is that the carry, if a 1, is added to the sum of the two upper bytes that are added after the lower byte, so that the result of adding the two 16-bit numbers is another 16-bit number.

The use of only 8 bits is called *single-precision* arithmetic. Where a number is stored as two bytes, the term is *double precision*. If the programmer thinks that he might be dealing with numbers that may be too large to store in two bytes, he may use three bytes or more, each addition process being linked to the one previously done on the byte before it by the carry bit. (For this reason, the carry bit is sometimes called the *link*.) This is known as *multiple precision* arithmetic, but is very rarely used.

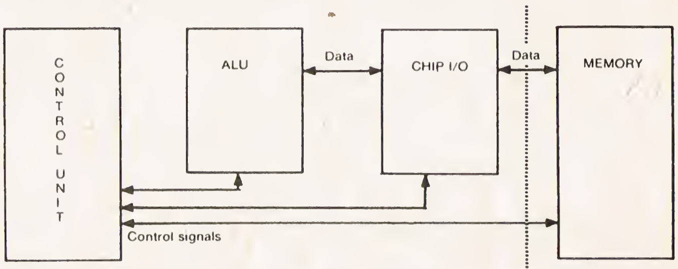


Fig. 3 — Data flow through microcomputer

Memory addressing

Every memory location can store 8 bits of data, and can be uniquely addressed by the bits put out by the computer on its address lines. (See next month's article on bus structure, peripherals, and I/O transfers). A memory address is purely a number assigned to a specific location so that the computer can remember where it stored things.

If we use an 8 bit data address, we can reference only 256 locations, which is hardly enough to be of any

practical value. Most microcomputers have a 16-bit program counter register (memory address of the next instruction to be executed), index register (used for referencing data in tables) and stack pointer (used for remembering where temporary values are stored) — all of which can be manipulated as two 8-bit registers and added as 16-bit numbers.

The next article in this series will show how the microprocessor interacts with its memory via the address and data lines, how peripheral devices are connected, and how transfers take place to peripherals using both interrupts and programmed transfers.

Last month reference was made to such terms as memory address, peripheral address, etc. We saw how an address can be represented as various combinations of bits, and how a 16-bit address has become a popular standard for both mini- and micro-computers, giving 65,536 available addresses, or 64K. ("K" is generally assumed to represent 1024 when talking in terms of memory size.

The Pigeon-Hole

It is sometimes difficult for the layman to understand the concept of memory addresses, and how the computer generates and the memory decodes them. The easiest way to visualise them is to think of a pigeon-hole system as would be used in a hotel for keeping messages for guests. Each box has a number assigned to it. When the desk clerk wants to leave a message for a guest, he simply puts in into a hole with the guest's room number marked beside the hole. Now let us assume that the pigeon-holes have only enough room for one message — i.e. when the clerk puts a second message into the hole, the first one falls behind the board and is lost. If he wants to read something from the last message for a guest, he simply goes to the location (room number) associated with that guest and retrieves the message. Naturally, if he wants to read the third last message that was put into the pigeon-hole he can't, because it has been lost.

An Electronic Pigeon-Hole

A memory system has the same kind of philosophy. You can store only one message (in a computer's case, a fixed number of bits) in any location. If you put another byte into that location, you will destroy the first. You can read it any number of times, however, because each time that you read it, it automatically gets put back into the same memory location for future use, until somebody writes a new byte into that location.

Now let's look at what hardware our system needs. The average MPU can address 65,536 bytes of memory, so it must have 16 lines coming from it, each being a 1 or 0, to indicate which of the 64K addresses it wants to use. Using a typical MPU chip, such as the 6800, we will need eight data bits to pass the data over from the MPU to the peripheral or vice versa.

We will also need some control signals to tell the peripherals (including memory) to look for their address, and what to do if they find it. Such signals are called "clock", "read", and "write".

SIGNALS USED	EQUIVALENT	BUS OPERATION
CONTROL	HEY!	CLOCK ALERTS ALL PERIPHERALS (INCLUDING MEMORY) TO LOOK FOR THEIR ADDRESS
ADDRESS	YOU!	ADDRESS LINES GIVE OUT A SPECIFIC ADDRESS
CONTROL & DATA	DO THIS!	DATA ARE EITHER TAKEN FROM OR PUT ONTO THE BUS LINES, DEPENDING ON THE STATUS OF READ/WRITE LINES

Memory Addressing

As a general convention, the address lines can be called A0 to A15, and the data lines will be called D0 to D7.

It is not necessary for every chip in a memory system to decode each one of the 16 address lines. What generally happens is that chips are arranged in convenient blocks, such as may be convenient to fill a particular size of memory board. In this case, logic circuitry on the board looks at all 16 bits (less the number required to address the individual locations on the board) and inhibits all chips on the board when the address signalled by the MPU is outside the range of that board.

A block of memory is assigned a starting address (the bus address of the lowest byte on that block) when the system is built. This is usually done with jumpers soldered to the board, or with DIP switches. In figure 2A, address bits A12 through A15 are assigned to determine the board number. As you can see, there are four bits, giving 16 combinations. Each of these combinations can have 4K addresses, which gives us our 64K (65,536) maximum addresses.

Let's assume that the memory has A12 to A15 set to all zeroes. Since we have specified that this board will be the lowest in memory, then it will respond. However, the chips on the board respond to 1024 addresses. If we just address this board and let the chips decode their 1024 addresses, we will get four chips all thinking that they are being addressed. To prevent this, we take the leftover bits A10 and A11 and use the four combinations of these two bits to select one of four groups of eight chips. Each chip is connected to one data line, the eight chips operating in parallel thus hold the eight bits of each byte.

Reading and Writing

We now see how the MPU can select a group of eight chips of the many on the bus. These selected chips will respond by either putting the eight bits that are stored in their addressed locations onto the bus, or by taking what is on the bus and storing it.

This can be controlled in many ways. Some systems have two separate signals, READ and WRITE. READ, when present, indicates to the peripheral that the MPU wants to input data, so it will put them onto the bus when selected. WRITE, when present, indicates to the peripheral that it should take the information from the bus and store it in its addressed location.

Which Way is Up

At this point a word of warning about READ and WRITE, INPUT and OUTPUT. To avoid the confusion that may arise because the MPU INPUTS while the peripheral is OUTPUTTING, a general convention exists which says that the words INPUT (or READ) and OUTPUT (or WRITE) are always used with respect to the MPU. In other words, OUTPUT data always flow from the MPU to a peripheral, during a WRITE operation, whereas INPUT data always flow from a peripheral to the MPU, during a READ operation. There is one exception to this rule, during NON-PROCESSOR TRANSFERS, which will be dealt with in the next article in this series.

ADDRESS RANGE	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
\$0000–03FF	2102	2102	2102	2102	2102	2102	2102	2102	ROW 0
\$0400–07FF	2102	2102	2102	2102	2102	2102	2102	2102	ROW 1
\$0800–0BFF	2102	2102	2102	2102	2102	2102	2102	2102	ROW 2
\$0C00–0FFF	2102	2102	2102	2102	2102	2102	2102	2102	ROW 3

Fig 2. Memory block addressing for a typical 4K × 8 board using 2102 (1K × 1) stataic RAMs. Board is set up as lowest of 16 possible places in a 64K system.

Fast Memory, Slow Memory, and Timing

There are two basic schools of thought on how a peripheral device such as memory should respond to a computer. One way is called a SYNCHRONOUS bus, the other is called ASYNCHRONOUS. Both have their advantages and disadvantages. A SYNCHRONOUS bus is one, such as is used in the 6800 MPU, in which a peripheral is given a precise amount of time to respond to a request from I/O. This time is fixed by the system clock. If the peripheral fails to respond properly during this time, the MPU carries on regardless, not knowing that its commands have not been carried out. Failure of a peripheral to do its job within the allocated time can result in totally unpredictable errors. For this reason, if a SYNCHRONOUS system has various memories attached to it, some very fast to respond, and some comparatively slow, the whole system will have to be slowed down so that the slowest device can operate reliably. The main advantage of this type of system is that it is far cheaper than the ASYNCHRONOUS bus. In the latter, a system is designed with various devices that are called MASTERS* and others that are called SLAVES. (For our present purposes, the MPU is the MASTER and the memory is the SLAVE). This system works in a HANDSHAKE fashion, i.e. one in which the MASTER, who controls the bus, sends out an address, a command, and a MASTER SYNC pulse. The MASTER SYNC pulse is similar to the clock pulse on a SYNCHRONOUS bus, except that as soon as MASTER SYNC is sent, the MASTER turns off its own internal clock. Thus, no more MASTER SYNC pulses are sent, and the bus sits in a state of limbo. When the peripheral is ready, it puts its information on the data bus (or takes information put there by the MASTER) and sends a signal called SLAVE SYNC. Upon receipt of SLAVE SYNC, the MASTER then carries on its work, normally issuing another MASTER SYNC to another address and so on. In this way, slow memories can take a long time to respond, while advantage can be taken of the extra speed of fast memories.

I/O to Slower Peripherals

So far, the only peripherals that have been mentioned are the various memory banks. Memory is fine, but it is so expensive (relatively) that it is only economically feasible to use it for data and programs that are currently being used, and to which access is very quickly needed (in the order of 1-2 uS). To store programs and data, we use such peripheral devices as tape drives (cassettes, 8-tracks, formatted etc.) discs (floppy, cartridge, multi-platter), paper tape, etc. For data that are not needed as quickly (in the order of 200-300 mS) such offline storage is useful because, even though it is extremely slow to get data from, it allows storage of from 200k bytes to many megabytes which would be prohibitively expensive using core or solid-state memory.

As an example of an I/O transfer, let's use a paper-tape punch on a model 33 teleprinter (TTY). That's about the slowest peripheral that you can get. I mentioned above that, in the case of an ASYNCHRONOUS bus, the MASTER can be made to wait while a SLAVE goes through its cycle. This is fine if the SLAVE delays the MASTER for a few hundred nanoseconds, or even a few microseconds, but the difference between that and the nine milliseconds required to send one byte to a TTY machine makes it very clumsy to hold up the bus for that long. Also, if the bus in question were to be SYNCHRONOUS, it means that the computer would have to be slowed down to fractions of a thousand times its normal speed — a ludicrous proposition.

What actually happens is that the data for punching are sent to a DATA REGISTER, which appears as a single memory address on the bus. Associated with each DATA REGISTER will be a STATUS REGISTER at another (usually the next sequential) memory address. The purpose of the STATUS REGISTER is to allow the MPU to monitor the progress of the data transfer to the slow device. Let's assume that bit 0 of the STATUS REGISTER indicates that the device which sends to the TTY is sitting idle. We can test for this condition in our program by doing a READ at the STATUS REGISTER address and seeing if this bit is set. We can then do a write of the data that we wish to send to the TTY at the DATA REGISTER address. The logic in the TTY interface will then clear bit 0 of the STATUS REGISTER and start sending the data, bit-by-bit, to the TTY. While this is happening, the MPU is free to address memory and other peripherals, and carry on executing a program. The program, for example, could be calculating an employee's paycheque, while the printer is printing that of the last employee to be processed. Such a program is called a FOREGROUND program. Every once in a while, the program can switch over to another program, called a BACKGROUND program, which checks to see if the PRINTER READY bit (bit 0) is set in the printer interface STATUS REGISTER, and if it is, sends the next data byte to the interface and switches the computer back into FOREGROUND mode. This can go on until the processor runs out of data to process, in which case it just keeps waiting for the printer interface to become idle before it can send another byte.

Why Just Sit There?

As you can see, the above way of doing things gives us the use of the processor while waiting for the peripherals, but we still have to waste time occasionally checking to see if the printer interface is ready. Wouldn't it be nice if the printer interface had some way of telling us when it's ready, so we don't have to keep checking? Well, such a system exists. This is called an INTERRUPT PROCESS, and together with DIRECT MEMORY ACCESS and the use of the console TTY, forms the content of the next article in this series. **CT**

*****_MASTER, SLAVE, MASTER SYNC, and SLAVE SYNC are terms and signals used on the DEC UNIBUS. They are used here as representative terms only._

TRITON Competition Result

The TRITON cross number competition, that appeared in the second of the Computing Today supplements that appeared in ETI, attracted a large number of entries. This fact plus the large number of correct answers pleased us as we had set the competition with the aim that it should be both fun to do and not require a Degree in Computer Science to complete. Judging by the response it seems as if we have met these goals.

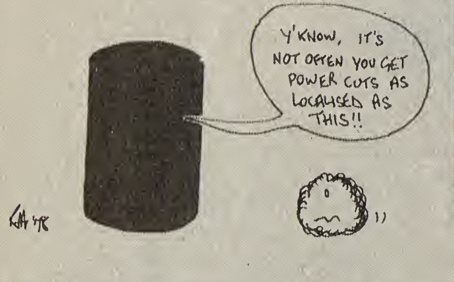
There could only be one winner and the lucky person's name was pulled out of the hat in early January. The winner was Mr. N. J. McAndrew of Colchester who was invited to the prize giving ceremony at Transam's Chapel Street shop a few days later. The bubbly flowed and the lucky Mr. McAndrew took his TRITON off into the pouring rain to begin construction in a few days off work courtesy British Rail. **CT**

Below Margaret Hewitt picks out the winner of the competition watched by Mike Hughes (foreground), designer of the Triton, and Gary Evans. Right, the winner receiving his Triton computer, the presentation being made by Nigel Stride and Graham Cliffton of Transam, the two gentlemen at the centre of the photograph.

(1) 8	0	(2) 8	0		(3) 3		(4) 1	
2		2		(5) 3	2	(6) 7	6	(7) 7
2		(8) 5	(9) 5	5		4		7
(10) 4	(11) 0	9	6		(12) 1	1	(13) 2	7
	0		(14) 0	0	7		1	
(15) 6	8	(16) 0	0		(17) 8	(18) 2	2	(19) 8
1		1		(20) 1	1	0		2
(21) 5	(22) 7	1	(23) 6	0		0		1
	4		4		(24) 2	1	0	2

computing today

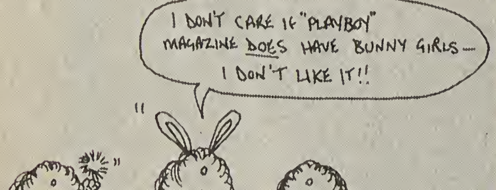
WHAT'S IN THE APRIL ISSUE . . . . . .



COMPUTER SURVEY

The number of small systems on the market has increased greatly over the past year and the choice of a machine to suit your application.

The April issue of Computing Today surveys some of the more popular small computers and presents in a clear, concise, fashion the capabilities and facilities offered by the different products.



CONSUMER SHOW

The recent Winter Consumer Electronic Show in Las Vegas saw the introduction of many new MPU based products including a chess challenger that talks.

Gerald Chevin was there for Computing Today and his report appears in the April issue.

NASCOM ADD ONS

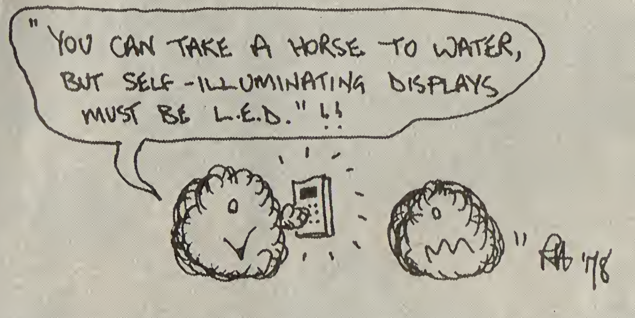
The NASCOM 1 computer has been one of the most successful of the DIY computer kits on the market recently NASCOM introduced a number of extras that allow the basic machine's potential to be considerably enhanced.

We take a look at the expansion board and RAM card as well as the TINY BASIC Nascom are now offering.

EXPANDA PET

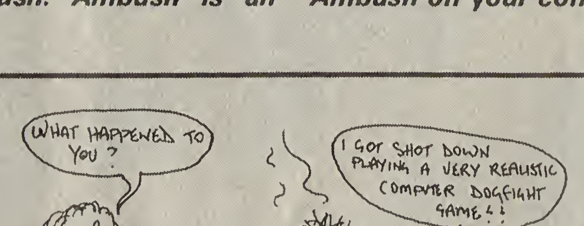
The commodore PET has been with us for over a year now but peripherals for the computer have been slow to appear. One of the essential devices in many applications is a floppy disk to provide a system of mass storage that is faster in operation than the tape system of the standard machine.

Next month we review the Compu/Think disk drive and diskmon operating system that will plug straight into your PET.



AMBUSH GAME

The April issue of our sister magazine, ETI, carries a project called Ambush. Ambush is an exciting space war game. Computing Today will carry a program that will allow those of you who don't dabble in electronics to play Ambush on your computer.



Plus all the regular features, news, softspot, hardlines and next month, a new regular letters page.

Beginning BASIC

Phil Cornes continues with our series introducing the BASIC language.

We hope you got on all right with last time's homework; some sample answers and another question are presented at the end of this article. By now, some of you must be thinking that it is all very well to be able to do vast amounts of calculation and decision making but as yet not a single answer has been printed out by the computer so that we can see the results of our labours. We will rectify this point straight away and go on to look at the main output of the BASIC language.

PRINT

The output statement of the BASIC language is the PRINT statement and an example of its use is given below—

```
10 Y = 1
20 A(Y) = Y*Y
25 PRINT A(Y)
30 IF Y = 5 THEN 9999
40 Y = Y + 1
50 GOTO 20
9999 END
```

This you should recognise as our $Y = X^2$ program from last time to which a PRINT statement has been added as line 25. Now line 20 calculates the value of A(Y).

The output from this program would be as follows—

```
1
4
9
16
25
```

each output being printed below the previous output in a vertical column.

We are going to spend some time on the PRINT statement as what you have just seen is the PRINT statement in only its simplest form, and it has several. Suppose, for example, that we wished to print a table with two columns, the first containing the value of X, the second the value of X^2 for values of X between 1 and 5. This is more or less what our program now does, except that values of X are not yet printed. If we replace line 25 with the following—

```
25 PRINT Y,A(Y)
```

then the output when the program was run would be—

```
1        1
2        4
3        9
4       16
5       25
```

which is just what we wanted. In effect, what happens is that the computer can split each output line on the VDU or teletype into 4 or 5 sections (depending on the number of characters per line) each about 16 characters long. Each of these is called a PRINT ZONE. Whenever a new line is begun on the output peripheral, the first item to be printed starts at the beginning of the first print zone on that line. When (as in our new line 25) a comma is encountered in a PRINT statement, it tells the computer that even though it has already output some data, there is more to follow on the same line so the computer will advance its cursor (the cursor points to the position on a line at which the next character will be printed) across the page to the beginning of the next totally empty print zone. The idea of "totally empty" print zone is brought in here because if you have print zones 16 characters wide and the output for the first print zone contains 18 characters (and so overflows into print zone 2) the next output will start at the beginning of print zone 3 as part of print zone 2 is already occupied.

There is another way of using the same idea as follows—

```
50 ........
60 Y = 2
70 PRINT Y,
80 Y = 3 + Y + Y
90 PRINT Y
100 .......
```

Notice here the comma after the "Y" in the print statement of line 70. As we have already said, this tells the computer that there is more data to follow on the same output line in the next empty print zone, so the output peripheral will wait while the calculation of line 80 is done then this result will be printed by the PRINT statement of line 90 alongside the first value of Y. One further point to note is that there is no comma following the "Y" in line 90, so if there is any subsequent output in the program it will now begin on a new line as the current line is finished with.

The next thing to note is that the PRINT statement has the ability to output messages as well as numerical answers so that, for example, you could get the computer to output— TODAY IS 'WEDNESDAY'

The actual PRINT statement necessary to achieve this would have the following format— 50 PRINT "TODAY IS 'WEDNESDAY' "

You should be able to see from this example that the message has been enclosed within inverted commas in the PRINT statement, and these tell the computer to output whatever is between them direct and not to try to find a numerical value for it. The only character which cannot be placed within the inverted commas for printing is the inverted comma itself. Therefore, if you wish to put a quote into a PRINT statement, you have to use the apostrophe instead, as in the example given.

Consider the following.

```
20 ........
30 X = 3*4/2
```

```
40 PRINT "THE VALUE OF 'X' IS",X
50 ........
```

Line 30 calculates a value for X, line 40 then goes on to PRINT the answer preceeded by the message (note the comma). The output looks like this—

```
THE VALUE OF 'X' IS          6
```

Notice the gap between the message and the answer. This arises because there are 19 characters in the message and with print zones 16 characters wide we just overflow into print zone 2 so the numerical value of X is printed in print zone 3. Ideally, we would like the output to appear as follows.

```
THE VALUE OF 'X' IS 6
```

and as you may have guessed this is possible on most machines by replacing the comma with a semi-colon—

```
40 PRINT "THE VALUE OF 'X' IS";X
```

The semi-colon has a similar effect to the comma in that it tells the computer that there is more output to follow on the same line, but the semi-colon differs in that it does not refer to print zones, but tells the computer to use close spacing between items to be printed (this can vary from 0 to 2 spaces depending on your machine).

FOR NEXT

It would be useful if there was an instruction in which we could state "execute this part of the program a number of times and then carry on with the rest of the program." Well (as you might have guessed from the sub-heading) there is such a statement in BASIC, the FOR NEXT statement which has the following general format.

```
FOR (variable) = (lower limit)TO(upper limit)STEP(increment)
NEXT(value of variable)
```

For example—

```
10 FOR Y = 2TO 6 STEP 1
20 ........
30 ........
40 ........
50 NEXT Y
60 ........
```

Here Y will take all values from 2 (lower limit) to 6 (upper limit) in steps of the increment (in this case, 1) so that the first time line 10 is executed, Y takes the value 2 and the program continues on until we reach line 50 (NEXT Y). We then go back to line 10 and STEP the variable by the increment and carry on to line 50 again. This looping continues until the value of the variable is greater than or equal to the value of the upper limit at which time the FOR NEXT loop is finished with. In our example, program execution would continue with line 60.

The lower limit, upper limit and increment can all be either constants, variables or expressions, so that—

```
FOR Q = A/B TO 19/C STEP R
NEXT Q
```

is a valid FOR NEXT statement.

One point worthy of note at this time is that there are two different ways of implementing a FOR NEXT loop on a computer, both of which are equally usable provided that you know which you have available.

Consider the following program—

```
10 FOR D = 2 TO 1 STEP 1
20 PRINT "TEST IN NEXT STATEMENT"
30 GOTO 9999
40 NEXT D
50 PRINT "TEST IN FOR STATEMENT"
9999 END
```

which statements will be executed in the running of this program? There are two possibilities. There are—

10 20 30 9999
or
10 50 9999

Why the difference? Well, you should see if you look back that at some point in the execution of a FOR NEXT loop, we ask the question "is the value of the variable greater than or equal to the upper limit". If it is, then we have finished with the loop; if it isn't, then we go round the loop again.

The difference in the two executions is dependant upon whether we ask this question in the FOR statement or the NEXT statement.

Assume we ask the question in the FOR statement, then line 10 will make D equal to the lower limit (in this case 2) and we then compare this with the upper limit and find that it is already greater. So we have finished with our FOR NEXT loop. Control then branches to line 50 to PRINT the message, and ENDs in line 9999.

If, on the other hand, we ask the question in the NEXT statement, then line 10 assigns the lower limit value of 2 to D and control passes to line 20 which forms part of the loop. Even if our test would fail, we go through the loop at least once before we find this out. Many writers consider it to be a 'bad' interpreter that operates in this manner (test in NEXT statement), but in practice I have never encountered any difficulties, and I have found that most of the computers I have used do operate this way.

There is one other point we need to consider about the last program, and that is line 30. It is quite permissible in BASIC to interrupt a FOR NEXT loop before it is completed by branching out of it to some other part of the program, but it is not permissible to branch into a FOR NEXT loop in such a way that the NEXT statement is encountered before the FOR statement: so that, if we were to add—

```
5 GOTO 40
```

to the above program, the computer would throw it out.

Another point to note is that FOR NEXT loops can be nested one within another. For example—

```
10 FOR X = 1 TO 5 STEP 1
20  FOR Y = 1 TO 3 STEP 1
30 PRINT X*Y,
40 NEXT Y
50 PRINT
60 NEXT X
70 END
```

(Note that the FOR NEXT statement in Y is completely enclosed by the FOR NEXT statement in X. This is known as nesting.)

If you were to run this program, you would find that it would produce the following output—

1	2	3
2	4	6
3	6	9
4	8	12
5	10	15

a simple multiplication table.

Initially, lines 10 and 20 set X and Y to 1, line 30 multiplies X and Y together and prints the result. (Notice the final comma in the PRINT statement.) We then jump back to line 20 and increase Y to 2 and print the new value of X*Y alongside the first. This is repeated for Y = 3. When we hit line 40 for the third time, it is ignored, and we go on to execute line 50. All this does, in effect, is to close the print statement of line 30 so that the next output will start on a new line.

Line 60 now takes us back to line 10, where X is increased to 2. Line 20 (when entered from above) now resets Y to 1, and the whole process is repeated with values of Y = 1; 2 and 3 again, and X = 2, producing a second line of output. The third, fourth and fifth lines of output are then produced in the same way using values of X of 3, 4 and 5 and then the program ends.

That's all for this month. Next month we go on to look at some new ways of assigning values to variables and some functions.

Here are possible answers to last times homework, and another question to be answered next month.

TAKE A CARD, ANY CARD

```
10 A = 1
20 R = RND(52)
30 A(A) = R
40 IF A = 52 THEN END
50 A = A + 1
60 GOTO 20
```

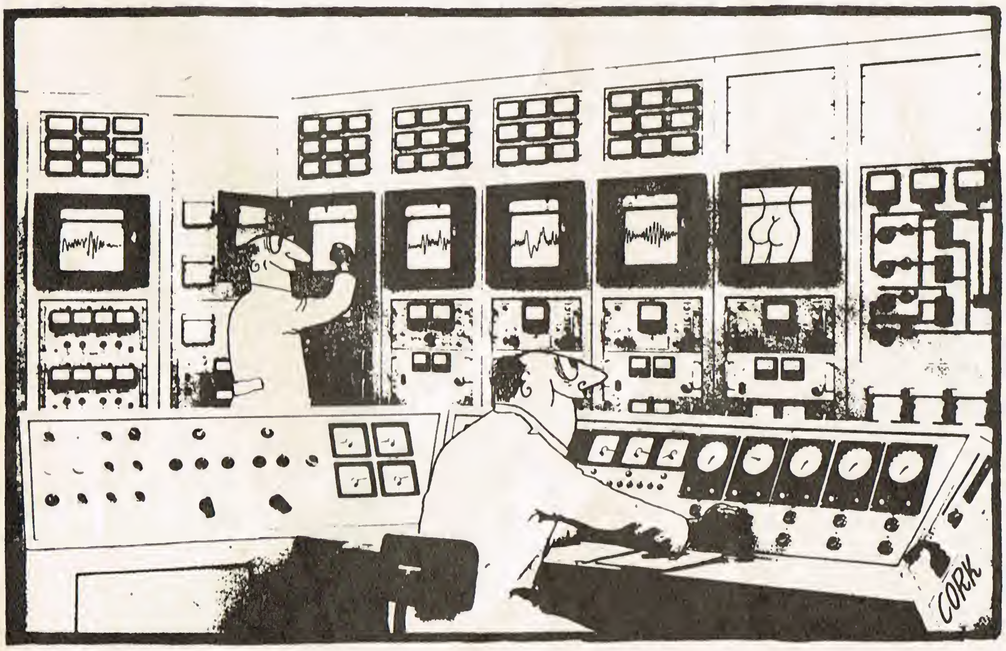
THE NEW ROUTINE

```
10 A = 1
20 R = RND(52)
30 B = 1
40 IF A(B) = R THEN 20
50 IF B = A THEN 80
60 B = B + 1
70 GOTO 40
80 A(A) = R
90 IF A = 52 THEN END
100 A = A + 1
110 GOTO 20
```

THE BRITISH SHUFFLE

```
10 A = 1
20 A(A) = A
30 IF A = 52 THEN 60
40 A = A + 1
50 GOTO 20
60 B = 1
70 R = RND(52)
80 X = A(R)
90 A(R) = A(B)
100 A(B) = X
110 IF B = 52 THEN END
120 B = B + 1
130 GOTO 70
```

And this month's homework is to introduce FOR NEXT loops into the above program (The British Shuffle) so as to eliminate some of the IF THEN loops, and also get the program to PRINT out the cards it generates. **CT**

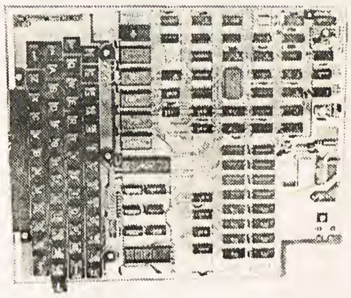


The Age of Affordable Personal Computing Has Finally Arrived

Ohio Scientifics

Superboard II

Full 8K basic and 4K user RAM
Built and tested



£263.84
+ 8% VAT

Ohio Scientific has made a major breakthrough in small computer technology which dramatically reduces the cost of personal computers. By use of custom LSI micro circuits, we have managed to put a complete ultra high performance computer and all necessary interfaces, including the keyboard and power supply, on a single printed circuit board. This new computer actually has more features and higher performance than some home or personal computers that are selling today for up to $2000. It is more powerful than computer systems which cost over $20,000 in the early 1970's.

This new machine can entertain your whole family with spectacular video games and cartoons, made possible by its ultra high resolution graphics and fast BASIC. It can help you with your personal finances and budget planning, made possible by its decimal arithmetic ability and cassette data storage capabilities. It can assist you in school or industry as an ultra powerful scientific calculator, made possible by its advanced scientific math functions and built-in "immediate" mode which allows complex problem solving without programming! This computer can actually entertain your children while it educates them in topics ranging from naming the President of the United States to tutoring trigonometry all possible by its fast extended BASIC graphics and data storage ability.

The machine can be economically expanded to assist in your business, remotely control your home, communicate with other computers and perform many of the other tasks via the broadest lines of expansion accessories in the microcomputer industry.

This machine is super easy to use because it communicates naturally in BASIC, an English-like programming language. So you can easily instruct it or program it to do whatever you want, *but you don't have to.* You don't because it comes with a complete software library on cassette including programmes for each application stated above. Ohio Scientific also offers you hundreds of inexpensive programs on ready-to-run cassettes. Program it yourself or just enjoy it, the choice is yours.

Standard Features

- Uses the ultra powerful 6502 microprocessor
- 8K Microsoft BASIC-in-ROM
- Full feature BASIC runs faster than currently available personal computers and all 8080-based business computesr.
- 4K static RAM on board expandable to 8K
- Full 53-key keyboard with upper-lower case and user programmability
- Kansas City standard audio cassette interface for high reliability
- Full machine code monitor and I/O utilities in ROM
- Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters, without overscan up to 30 x 30 characters.

Extras

- Available expander board features 24K static RAM (additional mini-floppy interface, port adapter for printer and modem and OSI 48 line expansion interface.
- Assembler/editor and extended machine code monitor available.

Commands

CONT	LIST	NEW	NULL	RUN	

Statements

CLEAR	DATA	DEF	DIM	END	FOR
GOTO	GOSUB	IF...GOTO	IF...THEN	INPUT	LET
NEXT	ON...GOTO	ON...GOSUB	POKE	PRINT	READ
REM	RESTORE	RETURN	STOP		

Expressions
Operators
—, +, *, /, ↑, NOT, AND, OR, >, <, <>, >=, <=, =
RANGE 10^{-32} to 10^{+32}

Functions

ABS(X)	ATN(X)	COS(X)	EXP(X)	FRE(X)	INT(X)
LOG(X)	PEEK(I)	POS(I)	RND(X)	SGN(X)	SIN(X)
SPC(I)	SQR(X)	TAB(I)	TAN(X)	USR(I)	

String Functions

ASC(X$)	CHR$(I)	FRE(X$)	LEFT$(X$,I)	LEN(X$)	MID$(X$,I,J)
RIGHT$(X$,I)			STR$(X)		VAL(X$)

Plus variables, arrays and good editing facilities.

Fully built and tested. Requires only +5V at 3 amps and a videomonitor or TV and RF converter to be up and running.

There is enormous interest in Superboard, so order early if you wish to avoid inevitably long delivery dates later this year.

FREE 15-DAY TRIAL
Lotus Sound have had so many enquiries with questions about various aspects of Superboard II that in order to save time and ensure your satisfaction we are offering to return the full price to anyone who returns their machine, in good order, within 15 days of delivery.

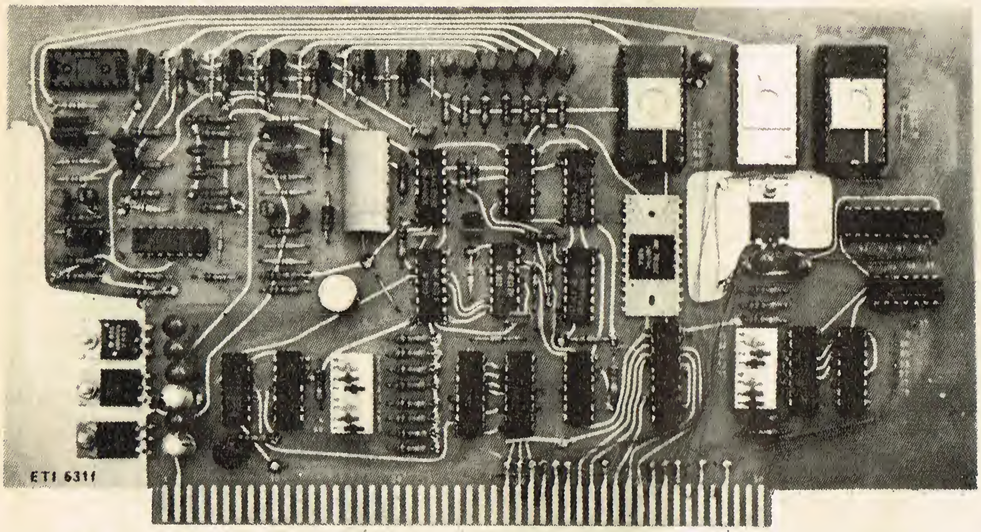
To: LOTUS SOUND
4 MORGAN ST., LONDON E3 5AB
Please send me ..... Ohio Scientific Superboard Computer(s)
I enclose cheque/PO for £ ....
Name ....
Address ....
....
....
CT3
....

SOUND

4 MORGAN ST., LONDON E3 5AB
(Phone for appointment)

PROJECT:
S100 Printer

Low cost hard copy is a reality with our S100 printer project.



While the cost of the large-scale integrated electronics built into computers has dropped, that of the mechanical peripheral devices has not followed this trend. Most printers cost several hundred, if not thousands, of dollars, so when we were shown a new mechanism which costs around £40 we were more than interested.

You rarely get something for nothing and this printer is no exception — it cannot do everything more sophisticated types can. It uses a 60mm wide paper which allows 32 characters per line, and the paper is a special metallized type. However it is still a very useful printer, especially for the hobbyist who doesn't have a grand to spare.

Design Features

As we seem to have standardised on the S100 bus this was the obvious choice for mechanical construction and electrical interfacing. People do offer an interface for the printer; however it requires the computer to be dedicated to it during the print cycle. The computer has to present and hold each character in sequence as »»

Specification

Print format	7 x 5 dot matrix
Number of different characters	127
Number of characters per line	32
Printing speed	2 lines per second
Character height	2.4 mm
Interface format	S100 bus compatible
Data entry time	5μs per character
Character storage capability	128
Power supply motor stopped	+16 V @ 100mA +8 V @ 350mA −16 V @ 80mA
motor running	+16 V @ 200mA +8 V @ 350mA −16 V @ 180mA
Printer mechanism	EUY−10E023LE (Datac)
Paper	EUY−SUB006

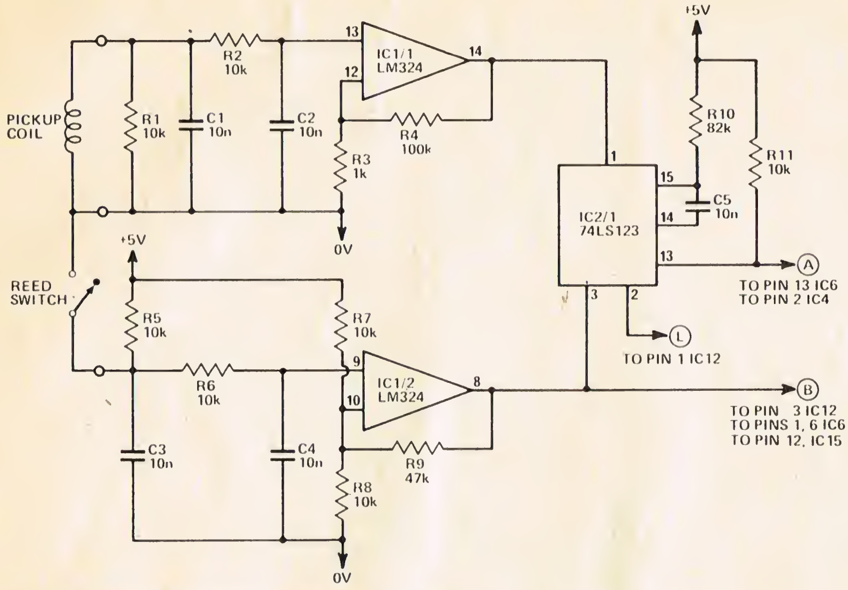


Fig. 1a. The circuit diagram of the pickup coil and reed switch buffer.

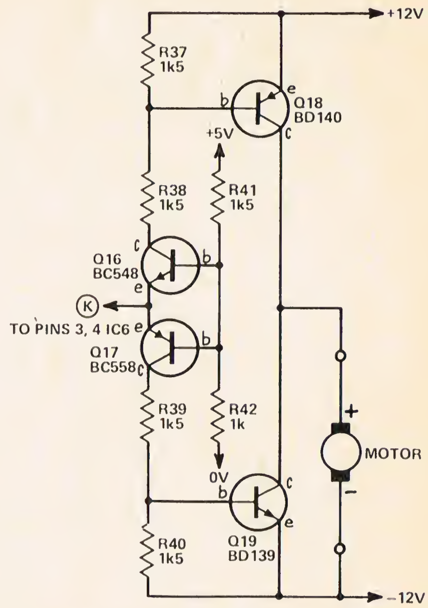


Fig. 1d. The motor drive interface.

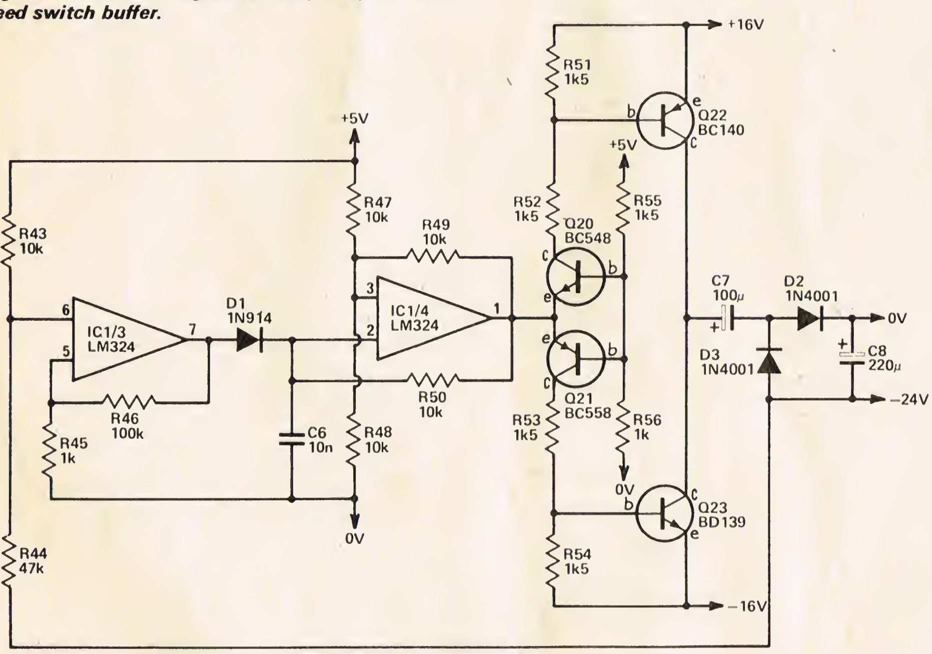


Fig. 1b. The — 24 volt power supply.

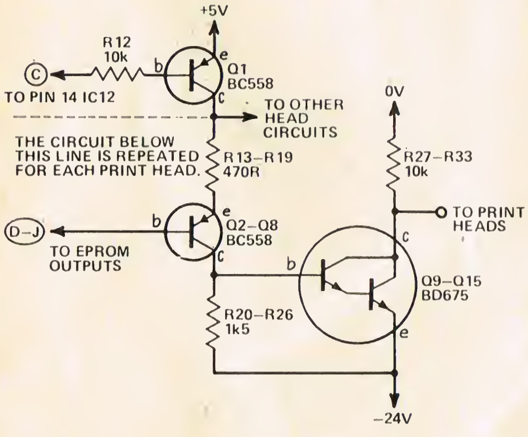


Fig. 1c. The circuit of the head drive. Although only one channel is shown there are 7 identical circuits.

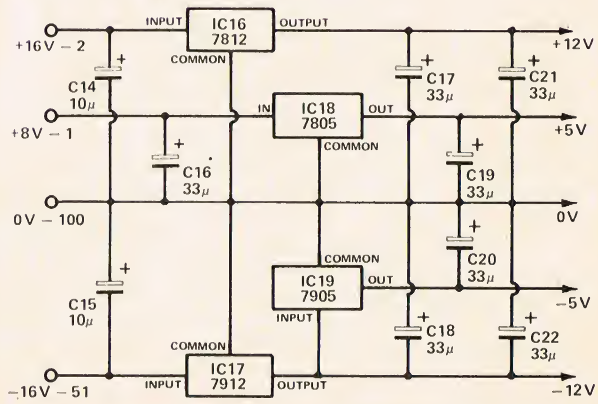


Fig. 1e. The Main power supply.

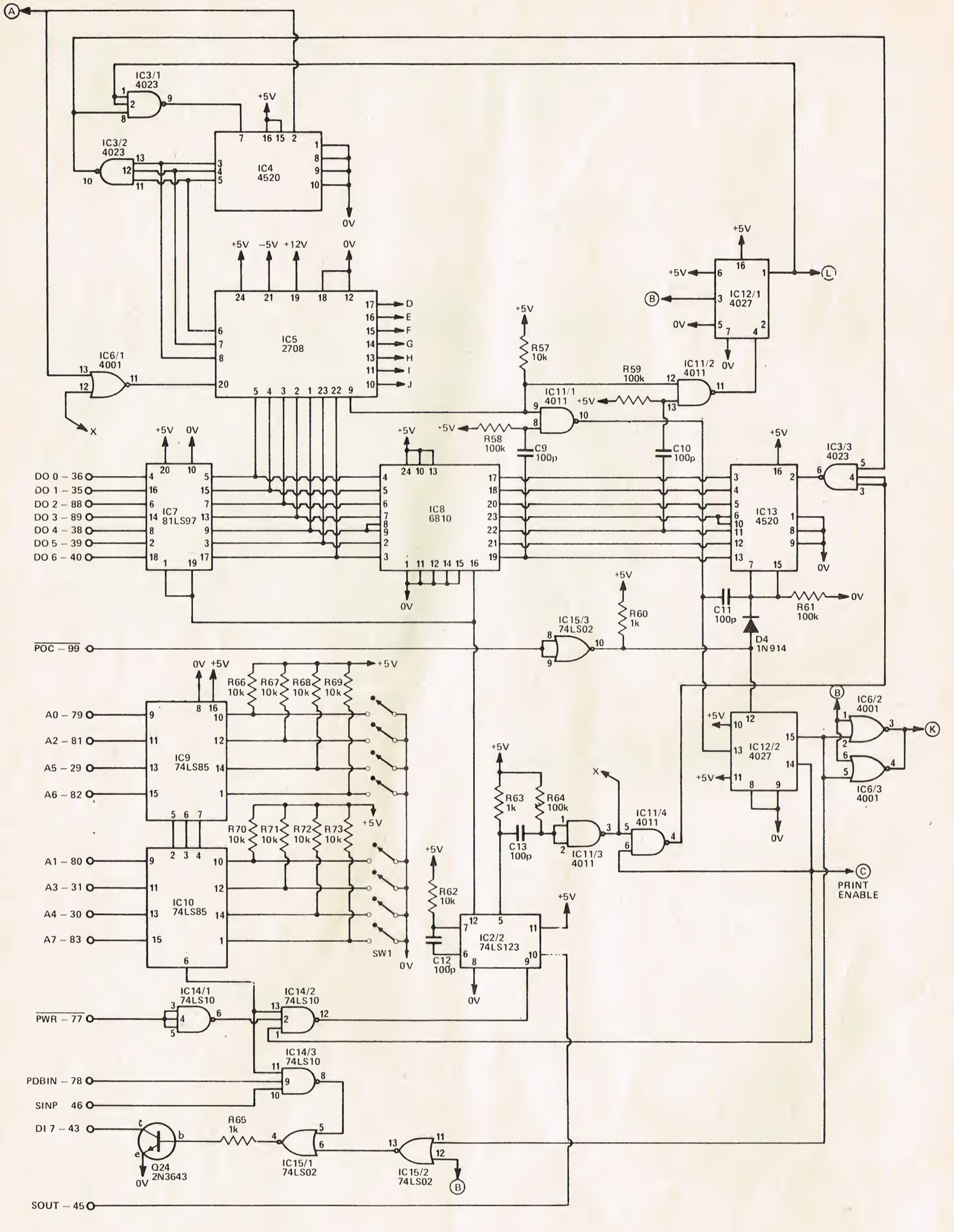


Fig. 1f. The main logic diagram.

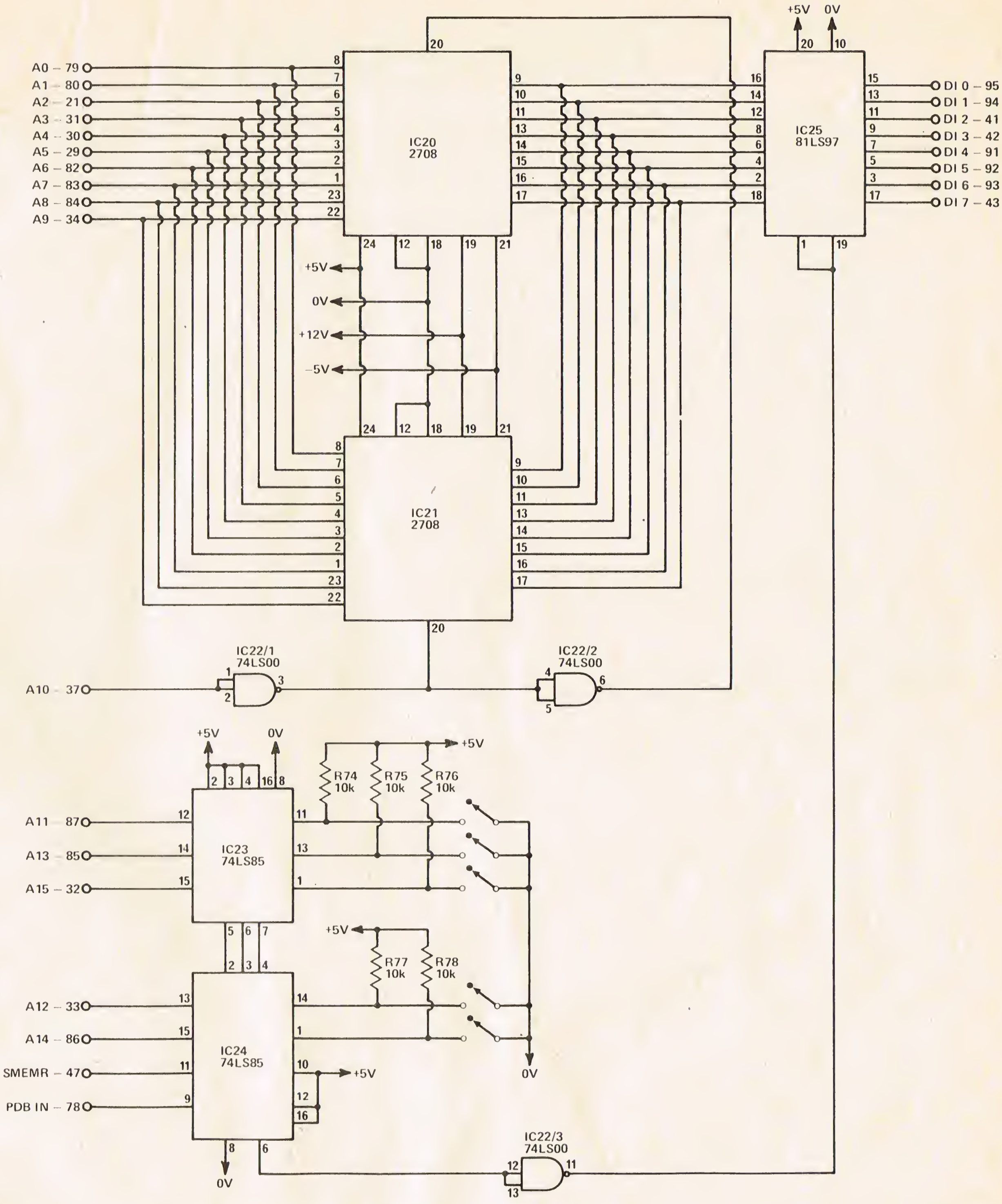


Fig. 1g. The circuit of the auxillary memory which can be used to store some of your software.

Buylines

The printer mechanism is available from Datac Limited at Tudor Road, Broadheath, Altrincham, WA14 5TN. We understand that Transam, 12 Chapel Street, London will offer a ready programmed character generator PROM. The rest of the components are fairly standard and should appear in the catalogues of most large mail order firms.

How It Works

Before starting an explanation of the electronics we will give a description of the printer used. It has a 24 Vdc motor which drives both the paper feed and the head drive. The head is simply a set of seven fine contacts arranged in a vertical line and is moved across the paper from left to right. The "paper" is metallized with a thin coating of aluminium, and by applying a voltage pulse between one of the head contacts and the paper the metallization is burnt off at that point. By applying pulses to each of the seven heads in the correct sequence as the head moves across the paper characters and words can be formed in a 5×7 dot matrix. The pulse required is -24 V for 240-480μs with a peak current of around 3 A per head.

At the end of the left to right scan of the head it returns quickly to the left while advancing the paper feed. The head is lifted off the paper on the return pass.

Also in the printer mechanism is a toothed wheel and pickup coil which gives an AC output of about 1 V which is used to synchronise the printing, and a reed switch which closes on the left to right passage of the print head. This is used to indicate the start of the line when printing.

The Electronics

The circuit is designed to operate on the S100 bus, and a proportion of the electronics forms an interface to the bus. The principle of operation of the unit is to present the data representing the first character to an I/O port along with the S100 timing signals to tell the printer circuitry to accept the character, and then repeat this process for up to 128 characters. No characters are printed until 128 characters have been output to the printer or until it recognises a carriage return. Printing starts immediately either of these events occurs and during printing a busy signal is available on the I/O port as no data can be entered while printing is in progress.

The S100 bus has available +8 V and ±16 V unregulated dc supplies, and from these we derive, using three-terminal regulators, both positive and negative 5 V and 12 V supplies. Also required for the printer is -24 V, and we derive this from the ±16 V supply using a diode pump type circuit. This consists of IC1/4 which is connected as a square wave oscillator running at 400 Hz. Its output drives the transistor buffer stage Q22-Q25 the output of which is a square wave of 32 V p-p. The capacitors C7 and C8, and the diodes D2 and D3 rectify this to give a negative voltage which if not limited would reach -30 V. However IC1/3 acts as a comparator and when the voltage on pin 6 drops below 0 V, which represents a voltage of -23.5 V, its output will go high, disabling IC1/4. This effectively regulates the -24 V supply.

Before we can print any data we must first store it. The data is presented to IC7 on the Data Out lines, then if the address presented to IC9 and IC10 is correct along with pin 10 (IC2) being high and a high pulse on pin 3 (IC14), the monostable IC2/2 is triggered. This produces a 500 ns wide pulse which enables the three-state buffer IC7, allowing the data to be written into the RAM IC8. At the end of this pulse, a second monostable (IC11/3) is triggered (about 5 μs) and during this time the contents of the EPROM are examined. If the character just written into the RAM is not a carrage return, pin 9 of that IC (IC9) will remain high. At the end of this 5 μs period, the address counter IC13 is incremented. The next character can now be entered.

If a carriage return is entered pin 9 of IC5 will low during this 5 μs wide pulse. This forces pin 10 of IC11 high resetting the address counter IC13 and clocking the flipflop IC12/2. If a carriage return is not detected but the 128th character has been entered pin 13 of IC13 will go low and this, via C9, will cause a positive pulse on the output of IC11/1 as well as causing the flipflop IC12/2 to be toggled.

Toggling this flipflop the first time causes pin 15 to go high and 14 low. This disables the monostable via IC14/2, and starts the motor. This is controlled by Q16-Q19; if point K is low Q16 and Q18 will turn on hard applying 24 V to the motor. When point K goes high, Q17 and Q19 will turn on, shorting out the motor and stopping it quickly.

Also reset by the carriage return is IC12/1, and a "0" will be applied to pins 1 and 2 of IC3 which holds IC4 reset. Once the motor starts, pulses are generated by the pickup coil. The output of the coil is filtered by R1, 2 and C1,2 to remove any high frequency interference and is then buffered by IC1/1 which is connected as a schmitt trigger. The output of IC1/1 is used to clock the monostable IC2/1 which generates the 350 μs wide pulse used for printing.

The reed switch is also filtered by R5,6 and C3,4 to remove contact bounce and noise, before being buffered by IC1/2 which is also connected as a schmitt trigger. The output of this IC is high from the start of the printing line until the start of the head return.

Once the print stroke has commenced the closing of the reed switch toggles the flipflop IC12/1, allowing IC4 to be clocked, IC4 then scans the 3 least significant address lines of IC5 on each successive clock pulse the EPROM is interrogated for 350 μs. The outputs from the EPROM are used to drive the print head circuitry.

After seven clock pulses IC3/2 detects this and resets IC4 back to zero so forming a divide by seven circuit. This pulse also clocks the RAM address counter IC13 to the next step. In this way, the RAM tells the EPROM what character it wants.

If a carriage return is detected the outputs of ICI1/1 and ICI1/2 will both go high, resetting IC12/1 preventing any further clocking of IC4. It also resets IC13 and clocks IC12/2 back to its original state where pin 14 is high and pin 15 low.

This allows data to be again entered, but as the reed switch is still closed the motor will continue to run due to the actio of IC6/72, 3 until the reed opens. If more than 32 characters were entered before the carriage return, after the first 32 characters have been printed pin 11 of IC13 will go low and the mono formed by C10/R59 causes IC12/1 to be reset, stopping IC4 from being clocked. IC12/2 however is not affected and the motor will continue to run, even after the reed switch opens. The printer then starts a second print stroke and the re-closing of the reed switch clocks IC12/1 allowing printing to continue.

The print head requires a negative 24 V pulse of 240-480 μs width with a peak current of about 3 A (for only 10 μs) while the metallization is evaporated. The drive consists of seven identical circuits each with an interface transistor and a drive darlington transistor. One additional transistor is used (Q1) to disable the print head while the EPROM is active during the write mode.

The carriage return detection is performed by the EPROM as part of its programming. As there are only seven heads but eight bits in the memory, the least significant bit is always programmed as "1" except for the carriage return character. While the CR character ψ is programmed in the EPROM it cannot be accessed on this printer.

The auxiliary EPROMs use a standard address decoding and buffering circuit and do not require explanation.

requested by the printer. This involves a fairly lengthy program (124 steps for the MEK 6800D2) as well as tying up the processor.

We therefore chose a different approach using a dedicated memory on the interface to store the characters which can be entered at any speed (up to approximately 5μs apart) until either 128 characters (the limit of the memory) or a carriage return has been transmitted. At this point the print cycle starts and no further action is required from the processor. We initially tested the unit using only a keyboard, entering data manually with the carriage return initiating printing.

This method simplifies the software required and only ties up the processor long enough for it to output data at its own rate.

As we had some space left over on the card we decided to make provision for two additional 2708 EPROMs and their associated decoding/buffering. These are completely independent of the printer logic and can be used to store any software the user wishes. We do use another 2708 as the character generator as we were unable to find a suitable commercial device at a reasonable price. As this EPROM has 1024 locations, using eight bytes per character, we can have 128 characters. We therefore chose the full upper and lower case font with some Greek and mathematical symbols thrown in for good measure. As we are limited to a 5×7 dot matrix character some of the lower case characters are a bit strange (the ones with tails normally below the line) but are still quite legible.

Construction

As this is an economical printer, it was decided that the expense of a

Parts List

RESISTORS
(All ½W 5%)

R1,2	10k
R3	1k
R4	100k
R5—R8	10k
R9	47k
R10	82k
R11,12	10k
R13—R19	470R
R20—R26	1k5
R27—R33	10k
R34—R36	Numbers not used
R37—R41	1k5
R43	1k
R43	10k
R44	47k
R45	1k
R46	100k
R47—R50	10k
R51—R55	1k5
R56	1k
R57	10k
R58,59	100k
R60	1k
R61	100k
R62	10k
R63	1k
R64	100k
R65	1k
R66—R73	10k
*R74—R78	10k

CAPACITORS

C1—C6	10n polyester
C7	100µ 25V electro
C8	220µ 35V electro
C9—C13	100p ceramic
C14,15	10µ 25V electro
C16—C22	33µ 16V tantalum

SEMICONDUCTORS

IC1	LM324 quad op-amp
IC2	74LS123 dual mono
IC3	4023 three input NAND
IC4	4520 dual ÷16
IC5	2708 8K EPROM
IC6	4001 two input NOR
IC7	81LS97 octal buffer
IC8	6810 128 × RAM
IC9,10	74LS85 comparitor
IC11	4011 two input NAND
IC12	4027 dual JK flipflop
IC13	4520 dual ÷16
IC14	74LS10 three input NAND
IC15	74LS02 two input NOR
IC16	7812 positive 12V reg.
IC17	7912 negative 12V reg.
IC18	7805 positive 5V reg.
IC19	7905 negative 5V reg.
*IC20,21	2708 8K EPROM
*IC22	74LS00 two input NAND
*IC23,24	74LS85 comparator
*IC25	81LS97 octal buffer
Q1—Q8	BC558
Q9—Q15	BD675
Q16	BC548
Q17	BC558
Q18	BD140
Q19	BD139
Q20	BC548
Q21	BC558
Q22	BD140
Q23	BD139
Q24	2N3643
D1	1N914
D2,3	1N4001
D4	1N914

MISCELLANEOUS
PC board
Datac printer EUY—10E023LE
Four 24 pin sockets
One 16 pin socket and header
Two 8 pole DIP switches
One 15 pin 0.156 inch edge connector

*These components are not required if the additional memory is not needed.

through-hole plated PCB was not warranted. This means that a lot of components are soldered on both sides of the board preventing the use of sockets except for the EPROMS and the 6810 RAM.

The board can be assembled with the aid of the overlay in fig. 2. If the additional EPROMs are not required these ICs and the associated components can be deleted. None of the components in this area are used as feedthroughs for the printer electronics. In the printer circuitry there are two links in the 0V rail and three more leading to the edge connector. If the additional EPROMs are used all the address and data lines are linked to the edge connector as it was not possible (without a plated-through board) to use copper tracks. We used thin enamelled wire of the type where the enamel will melt on soldering for all these links. The numbers on the PCB next to these points indicate the pins on the edge connector to which they are to be linked. Note that the connector is numbered 1-50 on the component side and 51-100 on the copper side.

Connection to the printer is made via a 16 pin IC socket using a piece of ribbon cable and a 16 pin DIP header. »»

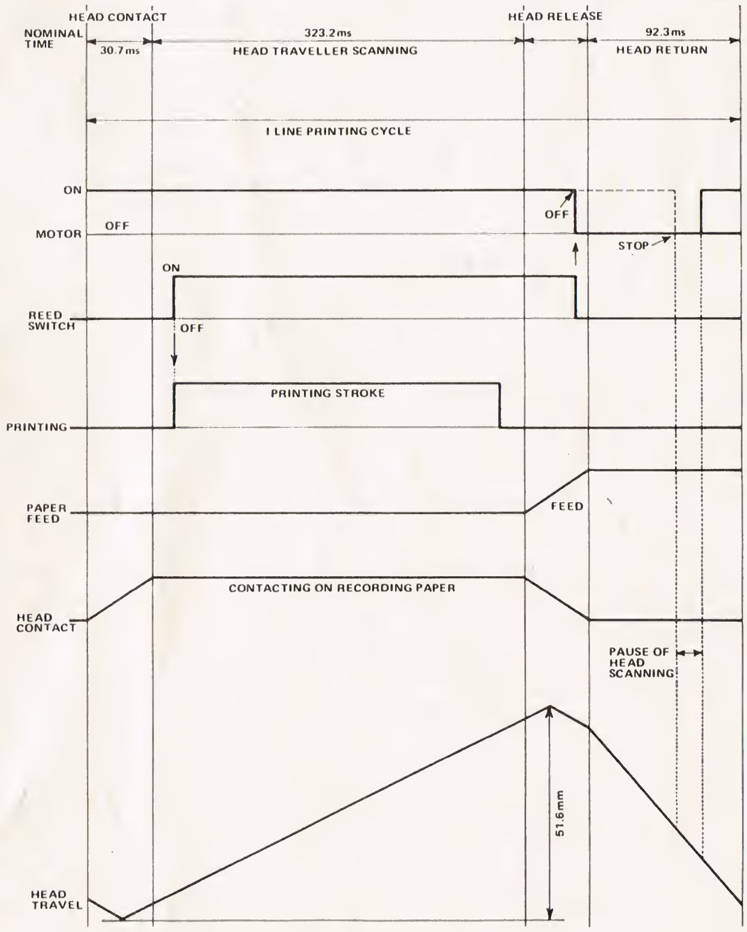


Fig. 3. The timing chart for the printer

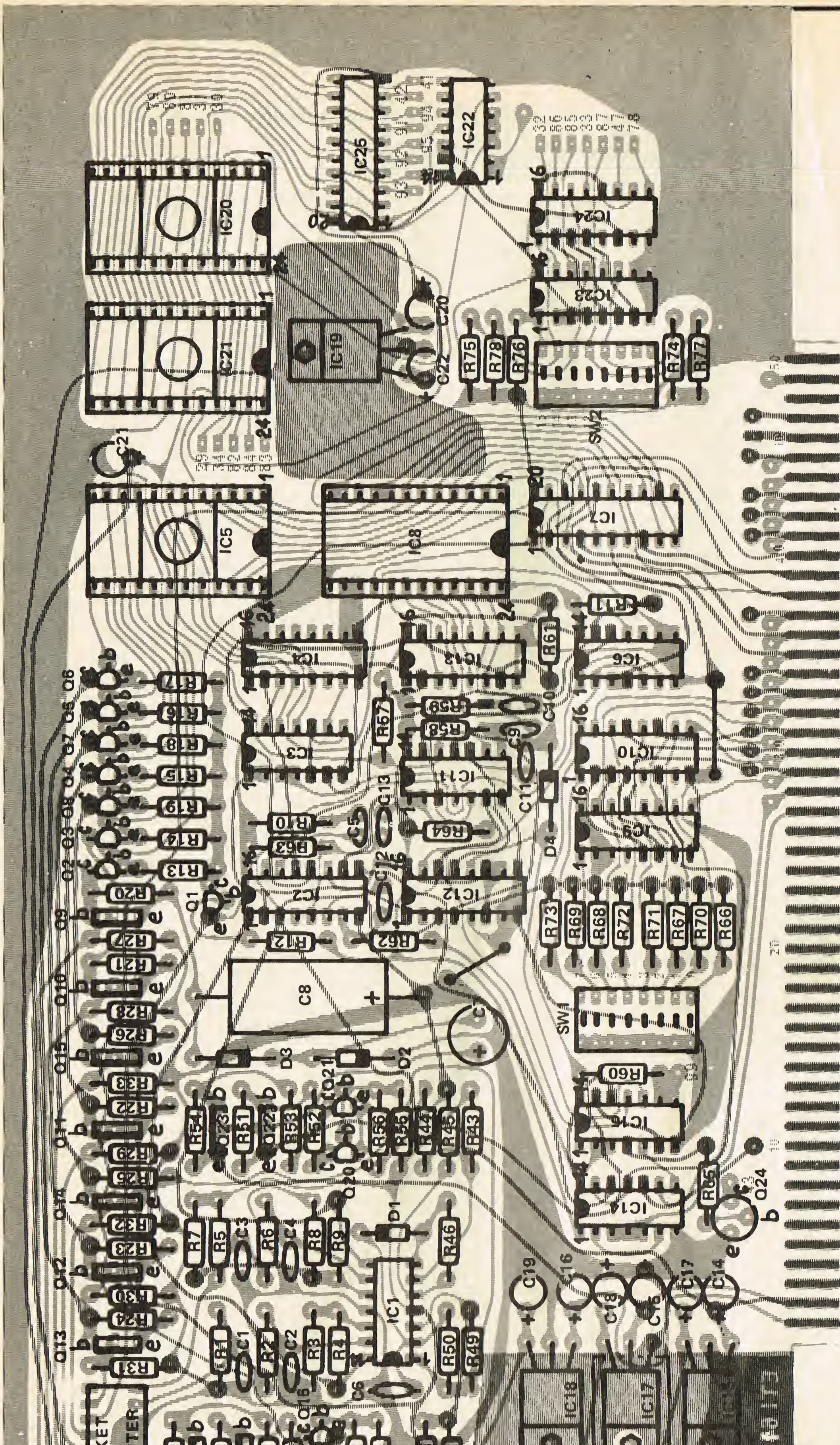


Fig. 2. The component overlay. Note that there must be an insulating washer under IC17 and that no additional heat sinks are needed on the voltage regulators.

CONNECTIONS

Header

Pin No.	
1	Head common
2	Motor –Ve
3	Motor +Ve
6	Coil-reed common
7	Pickup coil
8	Reed switch
10	Head 5
11	Head 4
12	Head 6
13	Head 3
14	Head 7
15	Head 2
16	Head 1

Printer

Pin No.	
1	Reed switch
2	Pickup coil
3	Coil-reed common
4	Motor +Ve
5	Motor –Ve
6	Head common
7	NC
8	Head 1
9	Head 2
10	Head 3
11	Head 4
12	Head 5
13	Head 6
14	Head 7
15	NC

Table 1

∝	000:000	377 363 355 355 363 355 377 377	@	002:000	377 203 175 105 125 207 377 377
β	000:010	377 377 201 155 155 223 377 377	A	002:010	377 201 157 157 157 201 377 377
γ	000:020	377 163 155 201 277 177 377 377	B	002:020	377 001 155 155 155 223 377 377
δ	000:030	377 223 155 155 163 377 377 377	C	002:030	377 203 175 175 175 273 377 377
ϵ	000:040	377 377 343 325 325 377 377 377	D	002:040	377 001 175 175 175 203 377 377
θ	000:050	377 203 155 155 155 203 377 377	E	002:050	377 001 155 155 155 175 377 377
ι	000:060	377 377 343 375 373 377 377 377	F	002:060	377 001 157 157 157 177 377 377
λ	000:070	377 335 353 367 373 375 377 377	G	002:070	377 203 175 175 155 141 377 377
μ	000:100	377 377 201 367 367 217 377 377	H	002:100	377 001 357 357 357 001 377 377
ν	000:110	377 337 301 373 367 317 377 377	I	002:110	377 377 175 001 175 377 377 377
π	000:120	377 357 301 337 301 337 377 377	J	002:120	377 373 175 175 003 177 377 377
Σ	000:130	377 175 071 105 155 175 377 377	K	002:130	377 001 357 327 273 175 377 377
Φ	000:140	377 347 333 001 333 347 377 377	L	002:140	377 001 375 375 375 375 377 377
ψ	000:150	376 316 366 000 366 316 376 376	M	002:150	377 001 277 317 277 001 377 377
ω	000:160	377 363 355 373 355 363 377 377	N	002:160	377 001 337 357 367 001 377 377
Ω	000:170	377 315 261 277 261 315 377 377	O	002:170	377 203 175 175 175 203 377 377
₀	000:200	377 377 363 355 355 363 377 377	P	002:200	377 001 157 157 157 237 377 377
₁	000:210	377 377 377 355 371 375 377 377	Q	002:210	377 203 175 165 173 205 377 377
₂	000:220	377 377 355 331 325 355 377 377	R	002:220	377 001 157 147 153 235 377 377
₃	000:230	377 377 333 335 325 353 377 377	S	002:230	377 235 155 155 155 163 377 377
⁰	000:240	377 377 237 157 157 237 377 377	T	002:240	377 177 177 001 177 177 377 377
²	000:250	377 377 267 147 127 267 377 377	U	002:250	377 003 375 375 375 003 377 377
±	000:260	377 333 333 213 333 333 377 377	V	002:260	377 007 373 375 373 007 377 377
÷	000:270	377 357 357 253 357 357 377 377	W	002:270	377 003 375 343 375 003 377 377
≈	000:300	377 267 157 267 333 267 377 377	X	002:300	377 071 327 357 327 071 377 377
√	000:310	377 367 373 001 177 177 377 377	Y	002:310	377 077 337 341 337 077 377 377
∫	000:320	377 373 375 203 177 277 377 377	Z	002:320	377 171 165 155 135 075 377 377
|	000:330	377 377 377 001 377 377 377 377	[	002:330	377 377 001 175 175 377 377 377
←	000:340	377 357 307 253 357 357 377 377	\	002:340	377 277 337 357 367 373 377 377
→	000:350	377 357 357 253 307 357 377 377	]	002:350	377 377 175 175 001 377 377 377
↑	000:360	377 357 337 203 337 357 377 377	^	002:360	377 337 277 177 277 337 377 377
↓	000:370	377 357 367 203 367 357 377 377	_	002:370	377 375 375 375 375 375 377 377
	001:000	377 377 377 377 377 377 377 377	`	003:000	377 377 177 277 337 377 377 377
!	001:010	377 377 377 015 377 377 377 377	a	003:010	377 373 325 325 325 341 377 377
"	001:020	377 377 037 377 037 377 377 377	b	003:020	377 001 355 355 355 363 377 377
#	001:030	377 327 001 327 001 327 377 377	c	003:030	377 377 343 335 335 335 377 377
$	001:040	377 333 253 001 253 267 377 377	d	003:040	377 363 355 355 355 001 377 377
%	001:050	377 073 067 357 331 271 377 377	e	003:050	377 377 343 325 325 347 377 377
&	001:060	377 363 215 145 233 365 377 377	f	003:060	377 377 357 201 157 277 377 377
'	001:070	377 377 337 277 177 377 377 377	g	003:070	377 377 357 327 325 343 377 377
(	001:100	377 377 307 273 175 377 377 377	h	003:100	377 377 001 357 357 361 377 377
)	001:110	377 377 175 273 307 377 377 377	i	003:110	377 377 355 241 375 377 377 377
*	001:120	377 327 357 203 357 327 377 377	j	003:120	377 377 377 375 243 377 377 377
+	001:130	377 357 357 203 357 357 377 377	k	003:130	377 377 001 367 353 335 377 377
,	001:140	377 377 375 363 377 377 377 377	l	003:140	377 377 175 001 375 377 377 377
–	001:150	377 357 357 357 357 357 377 377	m	003:150	377 341 337 347 337 341 377 377
.	001:160	377 377 377 371 377 377 377 377	n	003:160	377 377 341 337 337 341 377 377
/	001:170	377 373 367 357 337 277 377 377	o	003:170	377 377 343 335 335 343 377 377
0	001:200	377 377 203 175 175 203 377 377	p	003:200	377 377 301 327 327 357 377 377
1	001:210	377 377 275 001 375 377 377 377	q	003:210	377 357 327 327 301 375 377 377
2	001:220	377 275 171 165 155 235 377 377	r	003:220	377 377 301 357 337 337 377 377
3	001:230	377 173 175 135 055 163 377 377	s	003:230	377 357 325 325 325 373 377 377
4	001:240	377 347 327 267 001 367 377 377	t	003:240	377 377 337 203 335 377 377 377
5	001:250	377 033 135 135 135 143 377 377	u	003:250	377 303 375 375 301 375 377 377
6	001:260	377 303 255 155 155 363 377 377	v	003:260	377 307 373 375 373 307 377 377
7	001:270	377 177 161 157 137 077 377 377	w	003:270	377 303 375 363 375 303 377 377
8	001:300	377 223 155 155 155 223 377 377	x	003:300	377 335 353 367 353 335 377 377
9	001:310	377 237 155 155 153 207 377 377	y	003:310	377 377 317 367 365 303 377 377
:	001:320	377 377 377 311 377 377 377 377	z	003:320	377 335 331 325 315 335 377 377
;	001:330	377 377 375 311 377 377 377 377	{	003:330	377 357 357 223 175 175 377 377
<	001:340	377 357 327 273 175 377 377 377	¦	003:340	377 377 377 021 377 377 377 377
=	001:350	377 327 327 327 327 327 377 377	}	003:350	377 175 175 223 357 357 377 377
>	001:360	377 377 175 273 327 357 377 377	~	003:360	377 357 337 357 367 357 377 377
?	001:370	377 277 177 145 137 277 377 377	▒	003:370	377 125 253 125 253 125 377 377

Table 2

001:000	333	031		PRINT	IN	PRINTER	CHECK STATUS
001:002	376	177			CPI	177Q	BIT 7 ZERO?
001:004	312	000	001		JZ	PRINT	LOOP IF YES
001:007	176				MOV	A,M	FETCH CHAR
001:010	043				INX	H	NEXT CHAR
001:011	376	004			CPI	004Q	EOT?
001:013	312	023	001		JZ	END	END IF YES
001:016	323	031			OUT	PRINTER	OUTPUT CHAR
001:020	303	000	001		JMP	PRINT	ROUND AGAIN
001:023	076	015		END	MVI	A,015Q	LOAD CR
001:025	323	031			OUT	PRINTER	PRINT CR
001:027	311				RET		BACK TO CALLING ROUTINE

Using the Printer

The CT 641 Printer has been designed to interface most easily to S100-based computer systems, although it can be used with other bus structures, or even no bus at all. If it is plugged into an S100 system, the printer appears to the system to be a single I/O port. To print, the processor simply writes a string of characters in sequence to the output port, terminating with a CR character, whereupon the printer will itself initiate the print cycle, freeing the processor from any housekeeping.

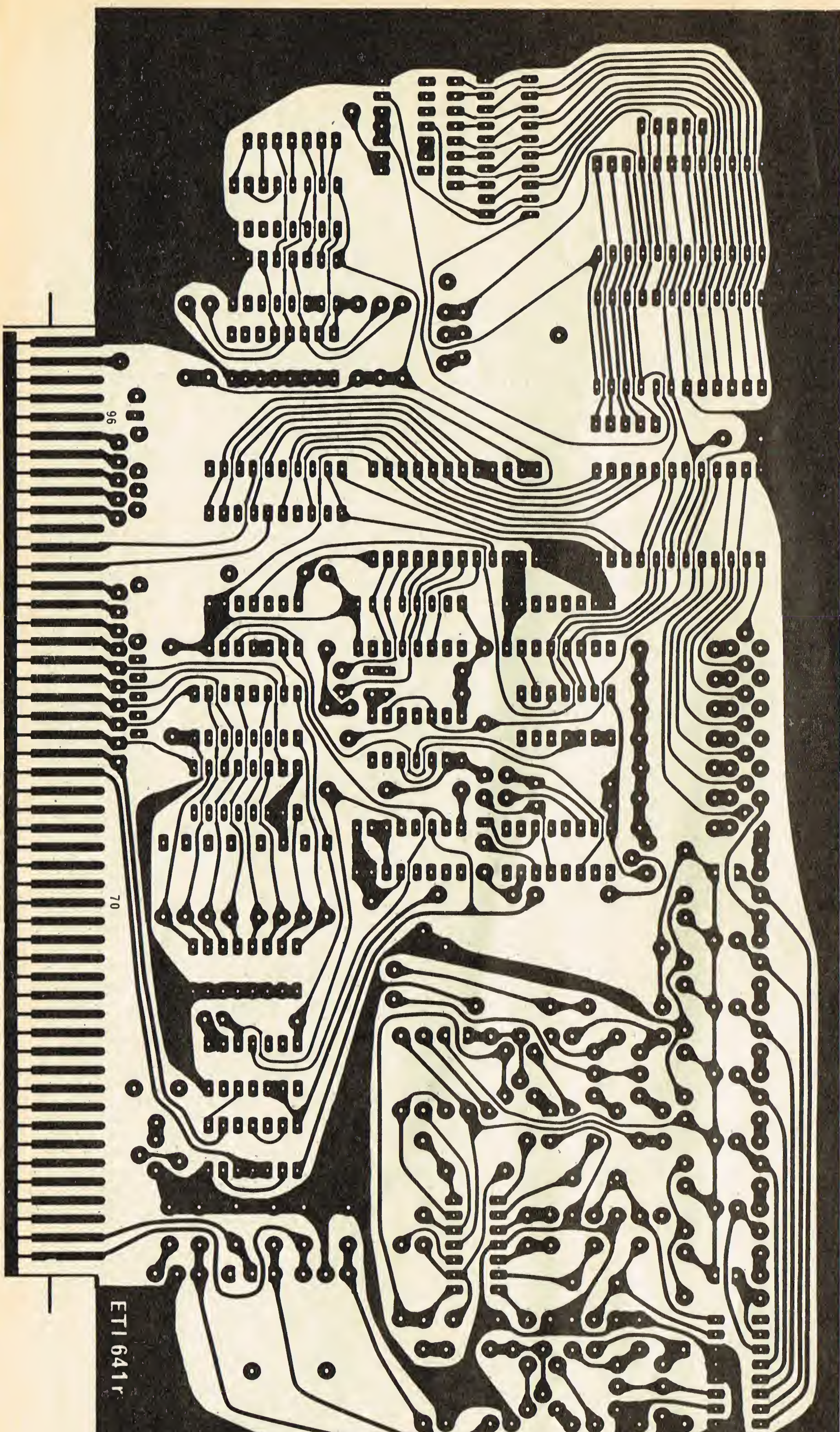
During the print cycle, however, the printer is unable to accept any characters, and signals this fact to the CPU by pulling bit 7 of the input port low for the duration of the print cycle (though this only appears when the input port is addressed). The printer driver routine should therefore check the status of the printer from this port before writing to it.

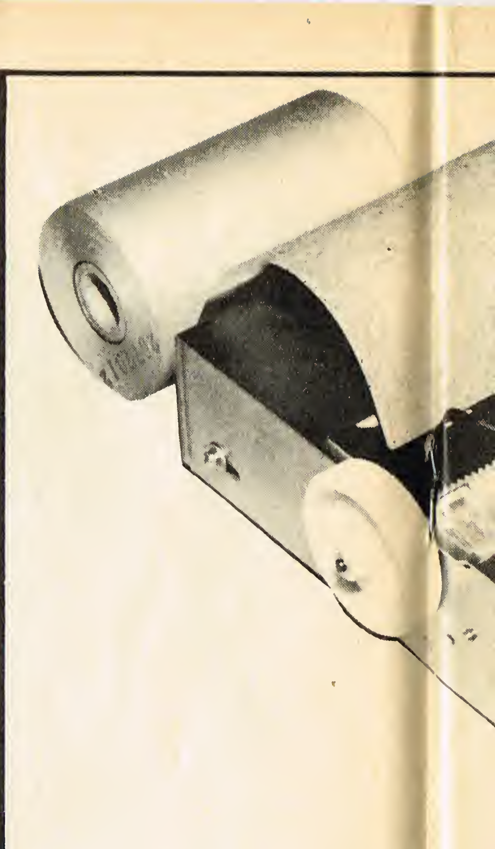
A general purpose printer driver which incorporates this feature is shown in Table 2. The calling program passes the starting address of the text to be output in the HL register pair, and the routine will then output all the text from there until it encounters an EOT character (004Q, 04H). When it finds an EOT, the routine substitutes a CR and outputs it to the printer to start the print cycle.

The printer input and output ports, although separate, share the same address, which can be set up on the 8-bit DIL switch SW1. In our example, the printer is set up for I/O address 031Q (19H). Although, the routine given is assembled at 001:000Q (0100H), it can easily be reassembled to any other address. Be sure when trying out the program, to initialise the Stack Pointer, as otherwise, the routine will return to 377:377Q and "gallop off into the wide blue yonder", possibly self-destructing for good measure.

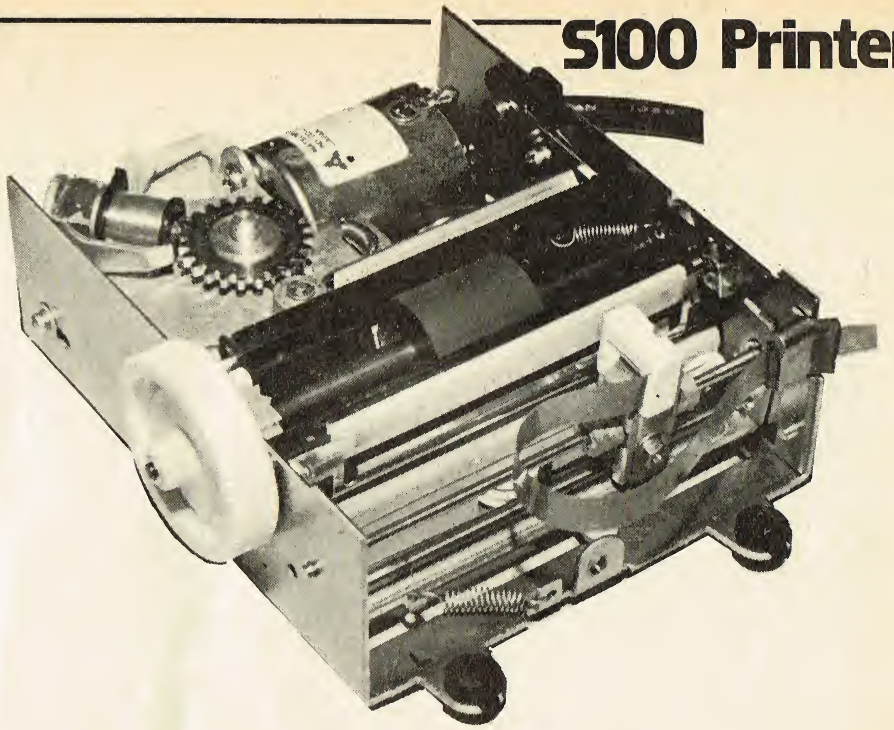
Another common trick used to indicate the end of message text is to set the most significant bit of the last character — as this is 7-bit ASCII it will not affect the printer or the CPU. However, the printer driver should recognise this and insert a CR, otherwise nothing will be printed.

The two EPROM sockets are addressed as a contiguous 2 K block of memory — they cannot be split apart. Consequently only 5 bits of address information have to be set on SW2 — one more bit selects which 2708 is addressed, and the final 10 bits are decoded inside the 2708s. »»

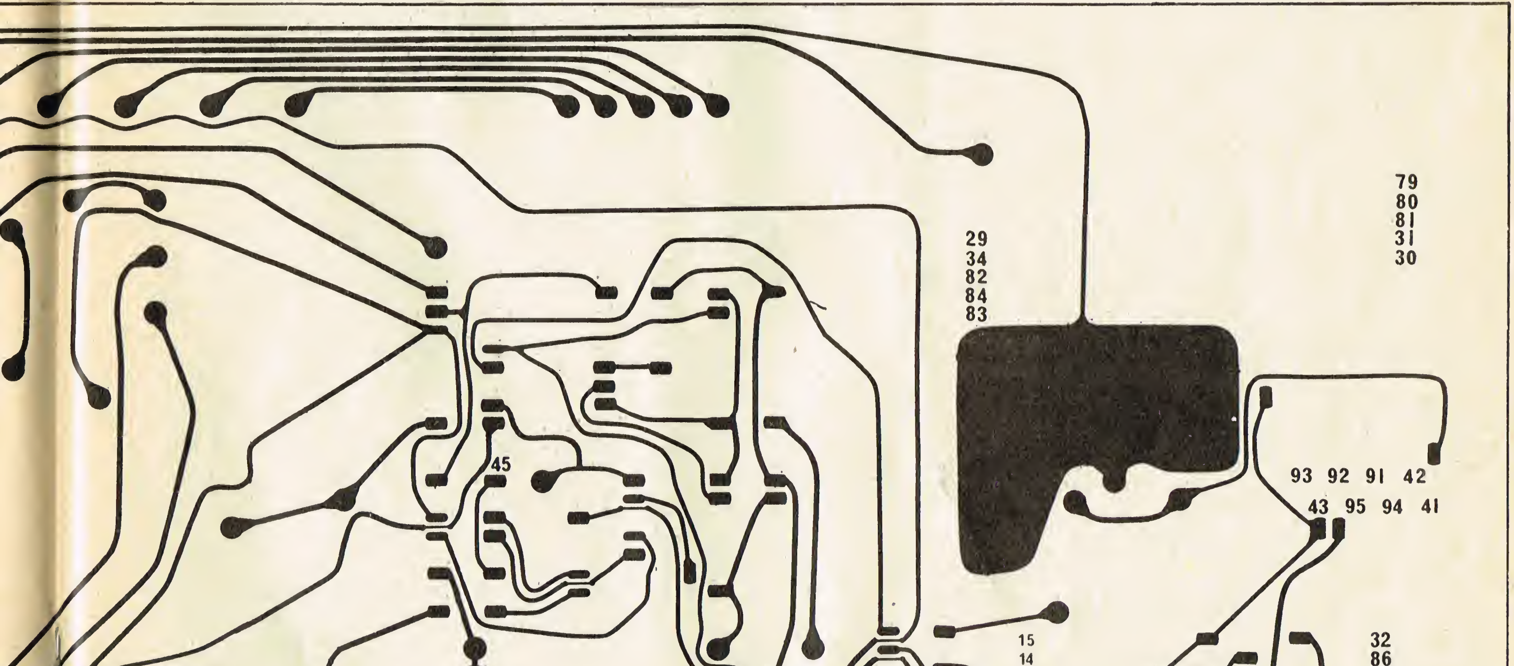

96
70
ETI 641r




ETI 641f
1
43



CT


79
80
81
31
30
29
34
82
84
83
45
93 92 91 42
43 95 94 41
15
14
13
12
11
7
6
5
4
3
2
1
0
99
32
86
85
33
87
47
78
43
10
20
30
40
50

computing today

technical book service

NEW:

Adams, C. BEGINNERS GUIDE TO COMPUTERS AND MICROPROCESSORS WITH PROJECTS £5.60
Understanding building programming and operating your own microcomputer.

NEW:

Albrecht, B. BASIC FOR HOME COMPUTERS. A self teaching guide £4.75
Shows you how to read, write and understand basic programming language used in the new personal size microcomputers.

Albrecht B. BASIC. A self teaching guide (2nd edition) £4.50
Teach yourself the programming language BASIC. You will learn how to use the computer as a tool in home or office and you will need no special maths or science background.

Alcock, D. ILLUSTRATING BASIC £2.20
This book presents a popular and widely available language called BASIC, and explains how to write simple programs.

Altman, I. MICROPROCESSORS £10.65
Gives a general overview of the technology design ideas and explains practical applications.

Altman, L. APPLYING MICROPROCESSORS £12.00
Follow volume which takes you into the second and third generation devices.

Aspinall, D. INTRO TO MICROPROCESSORS £6.40
Explains the characteristics of the component.

NEW:

Barden, W. Z-80 MICROCOMPUTER HANDBOOK £7.65

Barden, W. HOW TO BUY AND USE MINICOMPUTERS AND MICROCOMPUTERS £7.75
Discusses these smaller computers and shows how they can be used in a variety of practical and recreational tasks in the home or business.

Barden, W. HOW TO PROGRAM MICROCOMPUTERS £7.00
This book explains assembly language programming of microcomputers based on the Intel 8080, Motorola MC6800 and MOS Technology MCS6502 microprocessor.

Barna, A. INTRODUCTION TO MICROCOMPUTERS AND MICROPROCESSORS £7.50
Provides the basic knowledge required to understand microprocessor systems. Presents a fundamental discussion of many topics in both hardware and software.

Bibbero, R. J. MICROPROCESSORS IN INSTRUMENTS AND CONTROL £12.45
Introduces the background elements, paying particular regard to the dynamics and computational instrumentation required to accomplish real-time data processing tasks.

Lancaster, D. TV TYPEWRITER COOKBOOK £7.75
An in-depth coverage of tv typewriters (tv's) the only truly low cost microcomputer and small display interface.

Lancaster, D. CHEAP VIDEO COOKBOOK £6.50

Lesea, A. MICROPROCESSOR INTERFACING TECHNIQUES £7.50

NEW:

Leventhal. INTRO TO MICROPROCESSORS £16.70

NEW:

Lewis, T. G. MIND APPLIANCE HOME COMPUTER APPLICATIONS £4.75

NEW:

Libes, S. SMALL COMPUTER SYSTEMS HANDBOOK £5.75
The Primer written for those new to the field of personal home computers.

NEW:

Lippiatt. ARCHITECTURE OF SMALL COMPUTER SYSTEMS £4.35

NEW:

Moody, R. FIRST BOOK OF MICROCOMPUTERS £3.85
(the home computer owners best friend).

McGlynn, D. R. MICROPROCESSORS — Technology, Architecture & Applications £8.40
This introduction to the 'computer-on-a-chip' provides a clear explanation of the important new device.

McMurran, PROGRAMMING MICROPROCESSORS £5.50
A practical programming guide that includes architecture, arithmetic/logic operations, fixed and floating point computations, data exchange with peripheral devices computers and other programming aids.

NEW:

Nagin, P. BASIC WITH STYLE £4.00
Programming Proverbs. Principles of good programming with numerous examples to improve programming style and producing.

NEW:

Ogdin SOFTWARE DESIGN FOR MICROCOMPUTERS £7.00

NEW:

Ogdin. MICROCOMPUTER DESIGN £7.05

Peatman, J. B. MICROCOMPUTER BASED DESIGN £19.00

This book is intended for undergraduate courses on microprocessors.

NEW:
Bursky, D. MICROCOMPUTER BOARD DATA MANUAL £5.40
Bursky, D. MICROPROCESSOR DATA £5.40
Includes complete description of the processor. Support circuits, Architecture, Software, etc.
Coan, J. S. BASIC BASIC £7.50
An introduction to computer programming in BASIC language.
Coan, J. S. ADVANCED BASIC £5.30
Applications and problems.

NEW:
Duncan. MICROPROCESSOR SOFTWARE ENGINEERING £13.50

NEW:
Freiberger, S. CONSUMERS GUIDE TO PERSONAL COMPUTING AND MICROCOMPUTERS £5.50

NEW:
Frenzel, L. GETTING ACQUAINTED WITH MICROPROCESSORS £7.10
This is an invaluable book for those who want to know more about hobby and personal computing.
Gilmore, C. M. BEGINNERS GUIDE TO MICROPROCESSORS £4.75

Gosling, R. E. BEGINNING BASIC £3.25
Introduces BASIC to first time users.
Graham, N. MICROPROCESSOR PROGRAMMING FOR COMPUTER HOBBYISTS £7.00
Haviland, N. P. THE COMPULATOR BOOK £6.20
Building super calculators and minicomputer hardware with calculator chips.
Heiserman, D. L. MINIPROCESSORS FROM CALCULATORS TO COMPUTERS £4.85
Hilburn, J. L. MICROCOMPUTERS, MICROPROCESSORS, HARDWARE, SOFTWARE AND APPLICATIONS £16.95
Complete and practical introduction to the design, programming operation, uses and maintenance of modern microprocessors, their integrated circuits and other components.
Klingman, E. MICROPROCESSOR SYSTEMS DESIGN £16.95
Outstanding for its information on real microprocessors, this text is both an introduction and a detailed information source treating over a dozen processors, including new third generation devices. No prior knowledge of microprocessors or microelectronics is required for the reader.
Kemeny, J. G. BASIC PROGRAMMING £6.10
A basic text.
Korn, G. A. MICROPROCESSOR AND SMALL DIGITAL COMPUTER SYSTEMS FOR ENGINEERS AND SCIENTISTS £19.00
This book covers the types, languages, design software and applications of microprocessors.
Rao, G. U. MICROPROCESSOR AND MICROPROCESSOR SYSTEMS £20.50
A completely up-to-date report on the state-of-the-art of microprocessors and microcomputers written by one of the leading experts.
Rony, P. H. THE 8080A BUGBOOK: Microcomputer Interfacing & Programming £8.15
The principles, concepts and applications of an 8-bit microcomputer based on the 8080 microprocessor IU chip. The emphasis is on a computer as a controller.
Scelbi. 6800 SOFTWARE GOURMET GUIDE AND COOKBOOK £7.80
Scelbi. 8080 SOFTWARE GOURMET GUIDE AND COOKBOOK £7.80
Scelbi. UNDERSTANDING MICROCOMPUTERS £7.60
Gives the fundamental concepts of virtually all microcomputers.

NEW:
Schoman, K. THE BASIC WORKBOOK £3.70
Creative techniques for beginning programmers.

NEW:
Sirion, D. BASIC FROM THE GROUND UP £6.00
Soucek, B. MICROPROCESSORS AND MICROCOMPUTERS £18.80
Here is a description of the applications programming and interfacing techniques common to all microprocessors.

NEW:
Spracklen, D. SARGON £9.75
A computer chess program in Z-80 assembly language.

NEW:
Tracton. 57 PRACTICAL PROGRAMS & GAMES IN BASIC £6.40
Programs for everything from Space war games to Blackjack.
Waite. M. MICROCOMPUTER PRIMER £6.25
Introduces the beginner to the basic principles of the microcomputer.
Ward. MICROPROCESSOR / MICROPROGRAMMING HANDBOOK £6.00
Authoritative practical guide to microprocessor construction programming and applications.

NEW:
Veronis. MICROPROCESSOR £12.85
Zaks, R. INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING £7.50
Zaks, R. MICROPROCESSORS FROM CHIPS TO SYSTEMS £7.50

Note that all prices include postage and packing. Please make cheques, etc, payable to Computing Today Book Service (Payment in U.K. currency only please) and send to:
Computing Today Book Service,
P.O. Box 79,
Maidenhead, Berks.

Card Sharp

T. Lusty has written a games program that proves to be quite a memory test.

Most successful computer games are simulations of games or activities which people enjoyed before the computer existed. However, many card games seemed to have escaped the programmers. This seems a great shame as computer bridge might be just as enjoyable as computer chess.

Pelmanism is a card game which, some people claim, trains the mind to think and remember. The original game is not easily computerised but CARD-SHARP is a game which is easy and enjoyable to play and for which the same educational claims might be made.

CARD-SHARP is a game for two players which requires not only a good memory but also an intelligent strategy to obtain the best possible score. The first player (the computer) has a pack of playing cards face-up in front of him, and the second player is either blindfolded or in a position where he cannot see the cards. The first player then names either a suit or a card value and the second player suggests a card of the given suit or value. If the card has not been given before the second player scores points and the first player turns the given card face-down. If the suggested card is already face down the second player loses points and has to suggest another card which fits the first player's requirements. The game continues until all the cards are face-down.

The Pack of Cards

It is necessary for the computer to have some means of representing the pack of cards. The data structure must enable it to find easily which cards have already been used and to check that it does not ask impossible questions. The computer simulates the pack of array P which has dimensions 4 × 13. The array is initially set to zero to represent all the cards being face-up and then each element of the array is set to one as the required card is accepted. eg. If the value of P(3,11) is one, this would mean that you had already used the Jack of Hearts.

Two other array variables S and T are used to count the number of cards which have been given to the computer. The variable S with dimension 13 is used to count the number of cards of a particular value which the computer has received. eg. If the value of S(7) is three you have already given the computer 3 sevens. Variable T with dimension 4 is used to count the number of cards which have been given in a particular suit. eg. If the value of T(4) is 13 then you have already given the computer all the Spades.

Scoring

All good games must give the players a way of estimating how well they are doing, and the computer prints the score after every successful response. It has the decency not to gloat over how many points you have lost when your response was wrong!

The score is kept in variable K and points are added or subtracted depending upon whether or not the response is acceptable. Obviously, it gradually becomes more difficult to find a card which fits the computer's requirement and the scoring reflects this. If the computer asks for a card of a particular suit and receives an acceptable response the score is increased by one plus the number of cards already given in that suit; see line 810. If the computer receives an unacceptable response the score is debited by 20 minus the number of cards already given; see line 860. Similarly, if the computer asks for a card of a given value the score varies according to the prevailing situation; see lines 1090 and 1140.

Asking the Question

The computer's questions must obviously be unpredictable, but the random approach used here is not necessarily the best method. It must ask exactly 52 questions although some may be repeated when the response is incorrect, and there must be at least one possible answer.

The FOR . . . NEXT loop from line 570 to line 1180 is executed exactly 52 times and the NEXT statement can only be reached when an acceptable answer has been given. Within this loop the computer generates a random number X between 0 and 103 in line 600. If the number is greater than 51 a suit is required, if less, the computer asks for a card of a given value. There is, therefore, a 50-50 chance of either possibility occurring. If X is more than 51 it is divided by 4 and the remainder plus one gives the suit. (1=clubs, 2=diamonds, 3=hearts, 4=spades.) If X is less than 51 it is divided by 13 and the remainder plus one gives the value of the card required. (1=ace . . . 13=king.) Lines 640 and 920 check that there is a possible answer, and if there is not, the question is changed.

Input and Output

A possible reason why computers have remained a mystery to so many people for so long is their willingness to produce a hexadecimal dump rather than English. The quality of printout can vary greatly, from those programs which produce only the ubiquitous question mark to those which make witty comments after the input of rude words!

This program is fairly literate, if it asks for a spade it will accept 'ace' or 'an ace' as legitimate answers. If it requires a suit it accepts both the singular and the plural response. It also says 'please', but there is clearly room for improvement if you feel that way inclined.

The computer keeps its representation of the pack hidden away in memory, and the printout is designed so that it is not easy to look back for clues as to which cards have previously been played. If you want to see the cards you have already used, you can type 'status' but the computer then subtracts 10 from your score.

Running the Program

The program uses fairly standard BASIC and should run on most machines. The random number in line 600 might need some modification.

If you have a choice between working in half-duplex (ie. the printer or VDU responds directly to the keyboard) or full-duplex (ie. the computer is programmed to echo what is typed on the keyboard back to the display) then you may try the following inovation. Set up the processor to work in half-duplex and the terminal in full-duplex. Your input will not be printed and it is then impossible to cheat, the output from the program will not be affected.

The following is a listing of the source program, together with part of a sample run.

```
00100 PRINT "THIS IS A MEMORY TEST!!"
00110 PRINT "-------------------------------"
00120 PRINT
00130 PRINT
00140 PRINT "IMAGINE YOU HAVE A PACK OF CARDS.";
00150 PRINT "I SHALL ASK YOU TO GIVE ME A CARD"
00160 PRINT "EITHER 1) OF A GIVEN SUIT OR";
00170 PRINT "2) OF A GIVEN VALUE."
00180 PRINT
00190 PRINT "IF YOU GIVE ME A NEW CARD YOU SCORE MORE POINTS,";
00200 PRINT "BUT IF YOU TRY TO"
00210 PRINT "GIVE ME A CARD YOU HAVE ALREADY USED";
00220 PRINT "I SHALL TAKE POINTS OFF."
00230 PRINT
00240 PRINT "IF YOU WANT TO SEE THE CARDS YOU HAVE";
00250 PRINT "USED TYPE 'STATUS' IN REPLY"
00260 PRINT "TO ANY QUESTION, BUT YOU LOSE 10 POINTS FROM YOUR SCORE."
00270 PRINT
00280 PRINT
00290 PRINT "O.K. LET'S START."
00300 PRINT
00310 DIM P(4,13), N$(13), S$(4), T(4), S(13), M$(13), T$(4)
00320 REM ***** TURN ALL CARDS FACE-UP *****
00330 FOR I = 1 TO 4
00340 FOR J = 1 TO 13
00350 LET P(I,J) = 0
00360 NEXT J
00370 NEXT I
00380 REM ***** READ IN THE NAMES OF THE CARDS *****
00390 FOR I = 1 TO 13
00400 READ N$(I), M$(I)
00410 LET S(I) = 0
00420 NEXT I
00430 REM ***** READ IN THE SUIT NAMES *****
00440 FOR I = 1 TO 4
00450 READ S$(I), T$(I)
00460 LET T (I) = 0
00470 NEXT I
00480 REM ***** DATA FOR CARD NAMES *****
00490 DATA "ACE", "AN ACE", "TWO", "A TWO", "THREE", "A THREE", "FOUR"
00500 DATA "A FOUR", "FIVE", "A FIVE", "SIX", "A SIX", "SEVEN", "A SEVEN"
00510 DATA "EIGHT", "AN EIGHT", "NINE", "A NINE", "TEN", "A TEN", "JACK"
00520 DATA "A JACK", "QUEEN", "A QUEEN", "KING", "A KING"
00530 REM ***** DATA FOR SUIT NAMES *****
00540 DATA "CLUB", "CLUBS", "DIAMOND", "DIAMONDS"
00550 DATA "HEART", "HEARTS", "SPADE", "SPADES"
00560 REM ***** SET UP LOOP FOR 52 CARDS WITH COUNTER Z *****
00570 FOR Z = 1 TO 52
00580 REM ***** OBTAIN RANDOM NUMBER BETWWEN 0 AND 130 *****
00590 REM ***** AND CHOOSE TO ASK FOR CARD OR SUIT *****
00600 LET X = INT( 104*RND(0) )
00610 IF X<52 THEN 00910
00620 REM ***** SUIT CHOSEN --- X BETWEEN 52 AND 103 *****
00630 LET X = X-4*INT(X/4)+1
00640 IF T(X)>12 THEN 00630
00650 PRINT "PLEASE TYPE A"; S$(X); TAB(55);
00660 INPUT R$
00670 IF R$ <> "STATUS" THEN 00710
00680 GOSUB 01280
```

```
00690 GOTO 00650
00700 REM ***** CHECK TO SEE IF REPLY IS A VALID CARD *****
00710 FOR I = 1 TO 13
00720 IF R$ = N$(I) THEN 00790
00730 IF R$ = M$(I) THEN 00790
00740 NEXT I
00750 PRINT "DON'T BE STUPID!!";
00760 GOTO 00650
00770 REM ***** CHECK TO SEE IF THE CARD HAS *****
00780 REM ***** ALREADY BEEN GIVEN *****
00790 IF P(X,I) = 1 THEN 00860
00800 LET P(X,I) = 1
00810 LET K = K+T(X)+1
00820 LET S(I) = S(I)+1
00830 LET T(X) = T(X)+1
00840 PRINT "O.K. SCORE = "; K; TAB (21);
00850 GOTO 01180
00860 LET K = K-(20-T(X))
00870 PRINT
00880 PRINT "THINK AGAIN!!";
00890 GOTO 00650
00900 REM ***** CARD CHOSEN --- X BETWEEN 0 AND 51 *****
00910 LET X = X+1-13*(INT(X/13))
00920 IF S(X)>3 THEN 00910
00930 PRINT "PLEASE TYPE THE SUIT OF"; M$(X); TAB(55)
00940 INPUT R$
00950 IF R$ <> "STATUS" THEN 00990
00960 GOSUB 01280
00970 GOTO 00930
00980 REM ***** CHECK TO SEE IF REPLY IS A VALID SUIT *****
00990 FOR I = 1 TO 4
01000 IF R$ = S$(I) THEN 01070
01010 IF R$ = T$(I) THEN 01070
01020 NEXT I
01030 PRINT "DON'T BE STUPID!!";
01040 GOTO 00930
01050 REM ***** CHECK TO SEE IF THE CARD HAS *****
01060 REM ***** ALREADY BEEN GIVEN *****
01070 IF P(I,X) = 1 THEN 01140
01080 LET P(I,X) = 1
01090 LET K = K+S(X)+1
01100 LET S(X) = 2(X)+1
01110 LET T(I) = T(I)+1
01120 PRINT "OK. SCORE = "; K; TAB(21);
01130 GOTO 01180
01140 LET K = K-(10-S(X))
01150 PRINT
01160 PRINT "THINK AGAIN!!";
01170 GOTO 00930
01180 NEXT Z
01190 PRINT
01200 PRINT
01210 PRINT "YOU HAVE NOW GIVEN ME ALL 52 CARDS — THANK YOU."
01220 PRINT
01230 PRINT "YOUR FINAL SCORE IS"; K
01240 PRINT
01250 STOP
01260 REM $$$$$ SUBROUTINE TO PRINT CARDS THAT *****
01270 REM ***** HAVE ALREADY BEEN GIVEN *****
01280 PRINT
01290 PRINT
01300 PRINT T$(1), T$(2), T$(3), T$(4)
01310 PRINT "-----", "-----", "-----", "-----"
01320 PRINT
01330 FOR I = 1 TO 13
01340 FPR J = 1 TO 4
01350 IF P(J,I) = 0 THEN 01370
```

DIODES/ZENERS

QTY.				
	1N914	100v	10mA	.05
	1N4005	600v	1A	.08
	1N4007	1000v	1A	.15
	1N4148	75v	10mA	.05
	1N4733	5.1v	1 W Zener	.25
	1N753A	6.2v	500 mW Zener	.25
	1N758A	10v	"	.25
	1N759A	12v	"	.25
	1N5243	13v	"	.25
	1N5244B	14v	"	.25
	1N5245B	15v	"	.25

SOCKETS/BRIDGES

QTY.					
	8-pin	pcb	.20	ww	.35
	14-pin	pcb	.20	ww	.40
	16-pin	pcb	.20	ww	.40
	18-pin	pcb	.25	ww	.95
	20-pin	pcb	.35	ww	.95
	22-pin	pcb	.35	ww	.95
	24-pin	pcb	.35	ww	.95
	28-pin	pcb	.45	ww	1.25
	40-pin	pcb	.50	ww	1.25
	Molex pins	.01	To-3 Sockets		.25
	2 Amp Bridge		100-prv		.95
	25 Amp Bridge		200-prv		1.50

TRANSISTORS, LEDS, etc.

QTY.			
	2N2222	(2N2222 Plastic .10)	.15
	2N2222A		.19
	2N2907A	PNP	.19
	2N3906	PNP (Plastic Unmarked)	.10
	2N3904	NPN (Plastic Unmarked)	.10
	2N3054	NPN	.45
	2N3055	NPN 15A 60v	.60
	T1P125	PNP Darlington	1.95
	LED Green,	Red, Clear, Yellow	.15
	D.L.747	7 seg 5/8" High com-anode	1.95
	MAN72	7 seg com-anode (Red)	1.25
	MAN3610	7 seg com-anode (Orange)	1.25
	MAN82A	7 seg com-anode (Yellow)	1.25
	MAN74	7 seg com-cathode (Red)	1.50
	FND359	7 seg com-cathode (Red)	1.25

9000 SERIES

QTY.			QTY.		
	9301	.85		9322	.65
	9309	.35		9601	.20
	9316	1.10		9602	.45

MICRO'S, RAMS, CPU'S, E-PROMS

QTY.			QTY.		
	8T13	1.50		2107B-4	4.95
	8T23	1.50		2114	9.50
	8T24	2.00		2513	6.25
	8T97	1.00		2708	10.50
	74S188	3.00		2716 D.S.	34.00
	1488	1.25		2716 (5v)	59.00
	1489	1.25		2758 (5v)	23.95
	1702A	4.50		3242	10.50
	AM 9050	4.00		4116	11.50
				6800	13.95
	MM 5314	3.00		6850	7.95
	MM 5316	3.50		8080	7.50
	MM 5387	3.50		8212	2.75
	MM 5369	2.95		8214	4.95
	TR 1602B	3.95		8216	3.50
	UPD 414	4.95		8224	3.25
	Z 80 A	22.50		8228	6.00
	Z 80	17.50		8251	7.50
	Z 80 PIO	10.50		8253	18.50
	2102	1.45		8255	8.50
	2102L	1.75		TMS 4044	9.95

C MOS

QTY.		
	4000	.15
	4001	.15
	4002	.20
	4004	3.95
	4006	.95
	4007	.20
	4008	.75
	4009	.35
	4010	.35
	4011	.20
	4012	.20
	4013	.40
	4014	.75
	4015	.75
	4016	.35
	4017	.75
	4018	.75
	4019	.35
	4020	.85
	4021	.75
	4022	.75
	4023	.20
	4024	.75
	4025	.20
	4026	1.95
	4027	.35
	4028	.75
	4029	1.15
	4030	.30
	4033	1.50
	4034	2.45
	4035	.75
	4037	1.80
	4040	.75
	4041	.69
	4042	.65
	4043	.50
	4044	.65
	4046	1.25
	4048	.95
	4049	.45
	4050	.45
	4052	.75
	4053	.75
	4066	.55
	4069/74C04	.35
	4071	.25
	4081	.30
	4082	.30
	4507	.95
	4511	.95
	4512	1.10
	4515	2.95
	4519	.85
	4522	1.10
	4526	.95
	4528	1.10
	4529	.95
	MC14409	14.50
	MC14419	4.85
	74C151	1.50

LINEARS, REGULATORS, etc.

QTY.			QTY.			QTY.		
	MCT2	.95		LM323K	5.95		LM380 (8-14 Pin)	1.19
	8038	3.95		LM324	1.25		LM709 (8-14 Pin)	.35
	LM201	.75		LM339	.75		LM711	.45
	LM301	.45		7805 (340T5)	.95		LM723	.40
	LM308	.65		LM340T12	.95		LM725	2.50
	LM309H	.65		LM340T15	.95		LM739	1.50
	LM309K (340K-5)	1.50		LM340T18	.95		LM741 (8-14)	.35
	LM310	.85		LM340T24	.95		LM747	1.10
	LM311D	.75		LM340K12	1.25		LM1307	1.25
	LM318	1.75		LM340K15	1.25		LM1458	.65
	LM320H6	.79		LM340K18	1.25		LM3900	.50
	LM320H15	.79		LM340K24	1.25		LM75451	.65
	LM320H24	.79		LM373	2.95		NE555	.45
	7905 (LM320K5)	1.65		LM377	3.95		NE556	.85
	LM320K12	1.65		78L05	.75		NE565	.95
	LM320K24	1.65		78L12	.75		NE566	1.25
	LM320T5	1.65		78L15	.75		NE567	.95
	LM320T12	1.65		78M05	.75			
	LM320T15	1.65						

– T T L –

QTY.			QTY.			QTY.			QTY.		
	7400	.10		7482	.75		74221	1.00		74LS02	.30
	7401	.15		7483	.75		74367	.95		74LS04	.30
	7402	.15		7485	.55		75108A	.35		74LS05	.35
	7403	.15		7486	.25		75491	.50		74LS08	.35
	7404	.10		7489	1.05		75492	.50		74LS09	.35
	7405	.25		7490	.45		74H00	.15		74LS10	.35
	7406	.25		7491	.70		74H01	.20		74LS11	.35
	7407	.55		7492	.45		74H04	.20		74LS20	.30
	7408	.15		7493	.35		74H05	.20		74LS21	.35
	7409	.15		7494	.75		74H08	.35		74LS22	.35
	7410	.15		7495	.60		74H10	.35		74LS32	.35
	7411	.25		7496	.80		74H11	.25		74LS37	.35
	7412	.25		74100	1.15		74H15	.45		74LS38	.45
	7413	.25		74107	.25		74H20	.25		74LS40	.40
	7414	.75		74121	.35		74H21	.25		74LS42	.75
	7416	.25		74122	.55		74H22	.40		74LS51	.45
	7417	.40		74123	.35		74H30	.20		74LS74	.45
	7420	.15		74125	.45		74H40	.25		74LS76	.50
	7426	.25		74126	.35		74H50	.25		74LS86	.45
	7427	.25		74132	.75		74H51	.25		74LS90	.65
	7430	.15		74141	.90		74H52	.15		74LS93	.65
	7432	.20		74150	.85		74H53	.25		74LS107	.50
	7437	.20		74151	.65		74H55	.20		74LS123	1.20
	7438	.20		74153	.75		74H72	.35		74LS151	.85
	7440	.20		74154	.95		74H74	.35		74LS153	.85
	7441	1.15		74156	.70		74H101	.75		74LS157	.85
	7442	.45		74157	.65		74H103	.55		74LS160	.95
	7443	.45		74161	.55		74H106	.95		74LS164	1.20
	7444	.45		74163	.85		74L00	.25		74LS193	1.05
	7445	.65		74164	.60		74L02	.20		74LS195	.95
	7446	.70		74165	1.10		74L03	.25		74LS244	1.70
	7447	.70		74166	1.25		74L04	.30		74LS367	.95
	7448	.50		74175	.80		74L10	.20		74LS368	.95
	7450	.25		74176	.85		74L20	.35		74S00	.35
	7451	.25		74180	.55		74L30	.45		74S02	.35
	7453	.20		74181	2.25		74L47	1.95		74S03	.25
	7454	.25		74182	.75		74L51	.45		74S04	.25
	7460	.40		74190	1.25		74L55	.65		74S05	.35
	7470	.45		74191	1.25		74L72	.45		74S08	.35
	7472	.40		74192	.75		74L73	.40		74S10	.35
	7473	.25		74193	.85		74L74	.45		74S11	.35
	7474	.30		74194	.95		74L75	.85		74S20	.25
	7475	.35		74195	.95		74L93	.55		74S40	.20
	7476	.40		74196	.95		74L123	.85		74S50	.20
	7480	.55		74197	.95		74LS00	.30		74S51	.25
	7481	.75		74198	1.45		74LS01	.30		74S64	.15
										74S74	.35
										74S112	.60
										74S114	.65
										74S133	.40
										74S140	.55
										74S151	.30
										74S153	.35
										74S157	.75
										74S158	.30
										74S194	1.05
										74S257 (8123)	1.05
										8131	2.75

CABLE ADDRESS: ICUSD

TELEX #

HOURS: 9 A.M. - 6 P.M. MON. thru SUN.

INTEGRATED CIRCUITS UNLIMITED

7889 Clairemont Mesa Blvd. • San Diego, California 92111 U.S.A.

NO MINIMUM

COMMERCIAL AND MANUFACTURING ACCOUNTS INVITED
ALL PRICES IN U.S. DOLLARS. PLEASE ADD POSTAGE TO COVER METHOD OF SHIPPING.
ORDERS OVER $100 (U.S.) WILL BE SHIPPED AIR NO CHARGE.

PAYMENT SUBMITTED WITH ORDER SHOULD BE IN U.S. DOLLARS.
ALL IC'S PRIME/GUARANTEED ALL ORDERS SHIPPED SAME DAY RECEIVED.

CREDIT CARDS ACCEPTED:

Phone (714) 278-4394 BarclayCard / Access / American Express / BankAmericard / Visa / MasterCharge

SPECIAL DISCOUNTS

Total Order	Deduct
$35-$99	10%
$100-$300	15%
$301-$1000	20%

Asian Resource

We look at the first examples of Home Computers produced in the Far East.

CAST YOUR MIND back about how consumer electronic products have shifted their geographic origins from the US or Europe to the Far East. The transistor radio, cassette recorder, calculator, TV games and so on were developed originally in the countries each side of the North Atlantic but today we *expect* these products to come from Japan, Hong Kong, Taiwan or Korea. We mention Japan here because this is regarded as a major source but in the last few years the land of the Rising Sun has found it harder and harder to compete with their still-low wage neighbours.

The electronics manufacturing companies in the Far East are not known for their original design skills but once a market appears, they jump in with both feet and prices tumble. Many of the larger companies are already using MPU's in advanced TV games; they also have expertise in keyboards and printers so it is a natural step for them to start the production of home computers. There is a lot of evidence that within a year we'll be able to buy in Britain home computers carrying the 'Made in Hong Kong' label.

A report in the January 1979 issue of *Asian Sources Electronics* makes it clear that electronics companies in Hong Kong and Taiwan are showing a keen interest in entering this rapidly growing and profitable market.

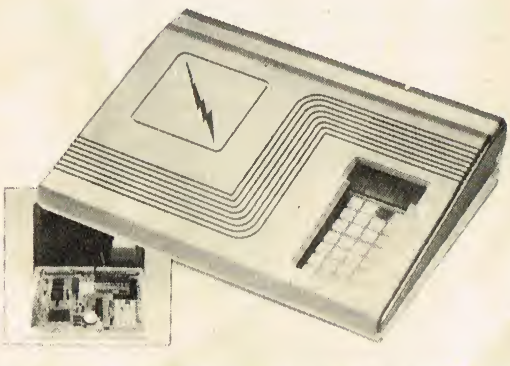
Few companies are actually producing at this moment. The American company *APF* are producing the *PeCos-I* in the Far East, the TV monitor being made in Hong Kong with the rest of the unit being made in Japan but high labour costs may lead to production being moved elsewhere in the area.

A Hong Kong company, *EACA Electronic Products Ltd*, are producing the *EG-2001 Video Genie* for an exclusive customer in the US. The system is S-100 bus compatible and comes with 4K ROM.

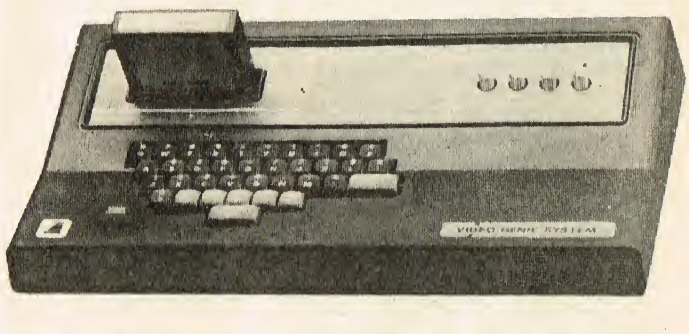
At least half a dozen other companies admit to some development work but many will also own up to the fact that they are weak on the necessary software. Far East companies have always fallen badly behind on instructions, whilst this is excusable with a transistor radio, the instructions are absolutely vital with any small computer system.

The PeCos One which is made in the Far East for a US customer. It includes 24K ROM and 16K RAM.

So far the very high technology necessary has proved a stumbling block but all the signs are that a number of home computers will soon be available. The implications are enormous. It was once thought that pocket calculators would always be a high-priced, small-production-run products but once manufacture started in the Far East, costs fell to fraction of even the most optimistic (pessimistic?) forecast. This in turn drove the original makers either into bankruptcy or into more and more sophisticated designs to keep ahead. This could happen in the home/small system computer field. CT



The Samson-1 is designed by the makers, Termbray Electronics Ltd of Hong Kong as a Microcomputer. This uses a 6502 with 4K ROM resident monitor and 1K of RAM; output is to a 6-digit HEX display.



The Hong Kong based EACA International Ltd have introduced the EG-2001 Video Genie for an exclusive US customer. Apart from being S-100 bus compatible and having 4K ROM, few details are available.

Microbiography: 6800 Series

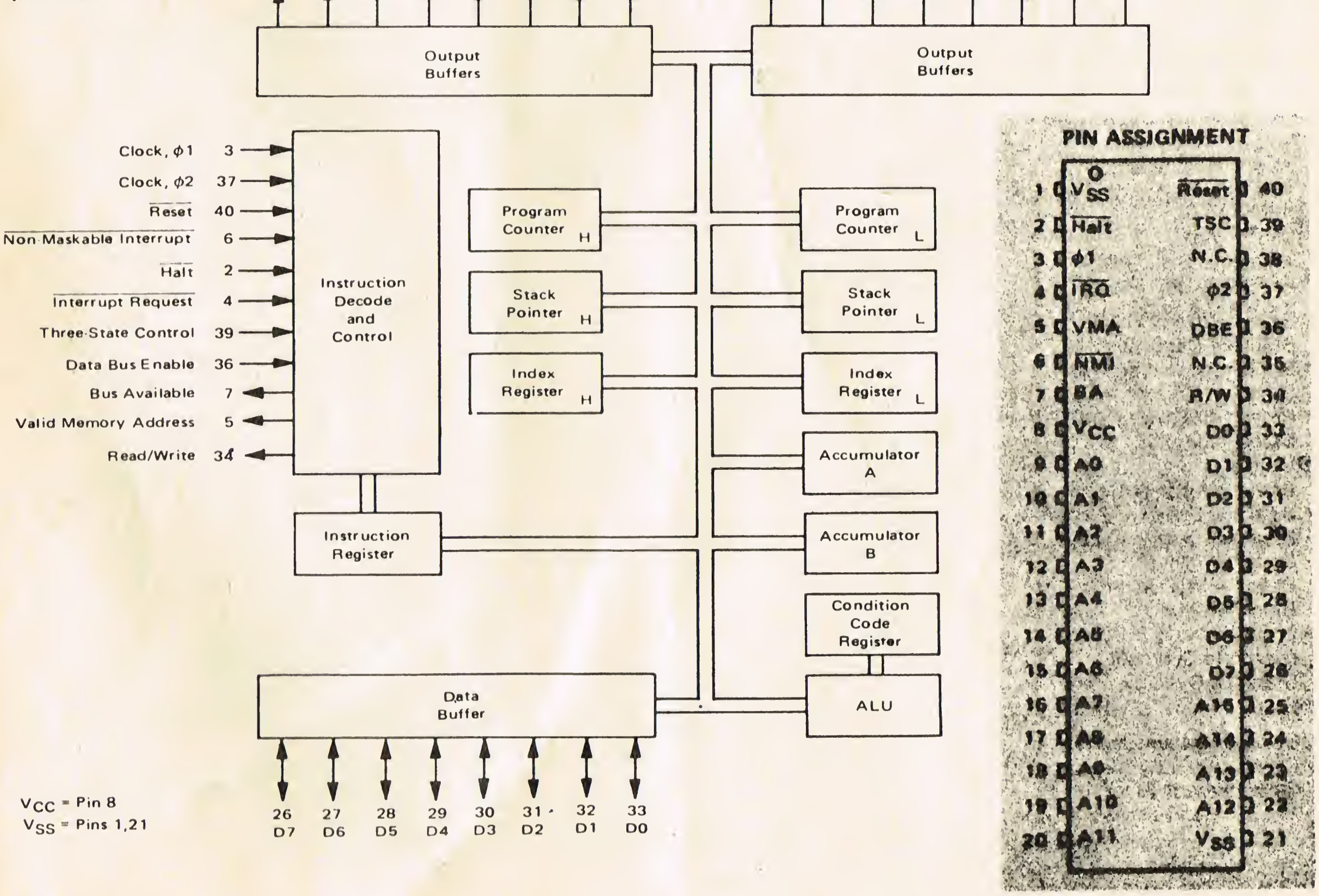
In the first installment of this series we have covered the 8000 series mpus and their assorted support chips. This month we move on to the "other" processor which is very popular in the hobbyist field, the Motorola 6800, and its close relatives in MOS Technology's 6500 series.

6800 hardware

The 6800 hardware is very straightforward, at least in block diagram form, figure 1. There are a number of registers shown attached to the internal bus, each with a specific function in mind, in contrast with the 8080, which includes several general purpose registers. Some of 6800 registers can however be used as general storage if not otherwise in use. To summarise: the two byte (16 bit) program counter of course keeps track of the current or next location in the program, and may be incremented, or otherwise altered in case of branches or subroutines. The stack pointer (16 bits) is used to record the next available space in the "Stack", an area in memory used for saving processor contents while performing subroutines, etc. The contents of the index register (16 bits) may be used to "offset" the address of the operand in index mode instructions (see below).

Two accumulators are included, these are the registers upon which the arithmetic and logical functions operate. The instruction register is used to hold the instruction while the instruction decoder is deciding what to do about it. Finally, the arithmetic logic unit (ALU) is the device which performs the arithmetic and logic functions. Associated with the ALU is the condition code register, whose individual bits indicate such conditions as zero, negative, carry, half-carry and interrupt. These conditions may be tested for conditional branches.

Fig 1. Block diagram of 6800 chip architecture and pinouts.



Interface to the rest of the microprocessor system is quite simple also. Sixteen bits of buffered address (1 TTL load, 139pF) and 8 bits of bidirectional data bus (same capability) are all present, plus an assortment of control inputs and outputs. A non-overlapping two phase clock with fairly strict requirements is used, for which purpose the MPQ6842 clock buffer is available. The other lines are as follows:
Three-State Control and Data Bus Enable: Inputs which control whether the address and data bus drivers are enabled. These allow possible external control of the buses by turning off the processor outputs (example — Direct Memory Access). They are rarely used for this purpose since almost all systems large enough to warrant this employ external buffers on the data and address buses, which are themselves controllable.
Reset: Initialises processor registers.
Interrupt Inputs: See below.
Halt: Causes processor to stop after current instruction completed, and all bus drivers turn off. May be used with some logic to achieve single instruction operation.
Bus Available: Indicates that the address bus is indeed available.
Valid Memory Address: Due to the internal workings of the mpu, miscellaneous signals may appear on the address outputs. To avoid inadvertently activating some unsuspecting device, VMA is used to tell every chip on the bus when to pay attention to the address, and when to ignore it.
Read/Write: Tells all devices whether to input or output to microprocessor. Power supply requirements are just 5V, and the original 1MHz model has been joined by 1.5 and 2MHz chips, with instructions taking 2 to 12 cycles.

Software

The instruction set for 6800 is shown in figure 2. For instructions involving an accumulator, either may be used. Seven addressing modes are available in various combinations.
Accumulator: One byte instructions operating on the accumulator.
Immediate: The second (or second and third for LDS and LDX) byte, following the op code, contains the operand.
Direct: The second byte of the instruction contains the address in zero page (lowest 256 bytes) of memory of the operand.
Extended: Second and third bytes contain the operand's address.
Indexed: The second byte is added to the index register, the result (held in a temporary register so as not to affect the index register) is used as the address of the operand. This mode is particularly useful for accessing tables, etc.
Implied: The instruction (one byte) applies to a particular internal register.
Relative Addressing: The second byte is used to branch forward or backwards from the current location up to +129 or −125.

The instruction set itself contains quite a comprehensive list of arithmetic and logic functions, branches with tests for all conditions, branch and jump to subroutine, plus manipulation of condition code bits. Versatility of the index register(s) and stack pointer is increased by several instructions.

Interrupts

In the 6800, four interrupt-like modes are possible. The first is RST which initialises the machine, and which causes a read from (hex) addresses FFFE, FFFF.

At these locations the processor fetches the address of the first routine to execute, presumably an initialisation routine.

During normal operation, a signal on the Non-Maskable Interrupt line will cause execution to stop after the current instruction, the next address is saved on the stack, and the address of the interrupt service routine is fetched from FFFC,D.

Operation of the Interrupt Request line is similar, except that by setting the interrupt mask bit in the condition code register, the mpu will ignore an IRQ signal. The IRQ address is FFF8,9. Finally there is an instruction, software interrupt (SWI) which causes an interrupt type action to take place, using addresses FFFA,B.

Fig 2. The instruction set of the 6800 mpu.

ABA	Add Accumulators
ADC	Add with Carry
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch if Carry Clear
BCS	Branch if Carry Set
BEQ	Branch if Equal to Zero
BGE	Branch if Greater or Equal Zero
BGT	Branch if Greater than Zero
BHI	Branch if Higher
BIT	Bit Test
BLE	Branch if Less or Equal
BLS	Branch if Lower or Same
BLT	Branch if Less than Zero
BMI	Branch if Minus
BNE	Branch if Not Equal to Zero
BPL	Branch if Plus
BRA	Branch Always
BSR	Branch to Subroutine
BVC	Branch if Overflow Clear
BVS	Branch if Overflow Set
CBA	Compare Accumulators
CLC	Clear Carry
CLI	Clear Interrupt Mask
CLR	Clear
CLV	Clear Overflow
CMP	Compare
COM	Complement
CPX	Compare Index Register
DAA	Decimal Adjust
DEC	Decrement
DES	Decrement Stack Pointer
DEX	Decrement Index Register
EOR	Exclusive OR
INC	Increment
INS	Increment Stack Pointer
INX	Increment Index Register
JMP	Jump
JSR	Jump to Subroutine
LDA	Load Accumulator
LDS	Load Stack Pointer
LDX	Load Index Register
LSR	Logical Shift Right
NEG	Negate
NOP	No Operation
ORA	Inclusive OR Accumulator
PSH	Push Data
PUL	Pull Data
ROL	Rotate Left
ROR	Rotate Right
RTI	Return from Interrupt
RTS	Return from Subroutine
SBA	Subtract Accumulators
SBC	Subtract with Carry
SEC	Set Carry
SEI	Set Interrupt Mask
SEV	Set Overflow
STA	Store Accumulator
STS	Store Stack Register
STX	Store Index Register
SUB	Subtract
SWI	Software Interrupt
TAB	Transfer Accumulators
TAP	Transfer Accumulators to Condition Code Reg.
TBA	Transfer Accumulators
TPA	Transfer Condition Code Reg. to Accumulator.
TST	Test
TSX	Transfer Stack Pointer to Index Register
TXS	Transfer Index Register to Stack Pointer
WAI	Wait for Interrupt

The 6800 interrupt mechanism means that interrupt servicing will generally be by "polling". That is to say, if several chips are all attached to an interrupt line, then the service routine "asks" each one in turn (by reading the device's status register) who interrupted. The order of asking determines the priority of each device. This is cheap but slow. Vectored interrupt capability is possible with complex, non standard hardware.

6500

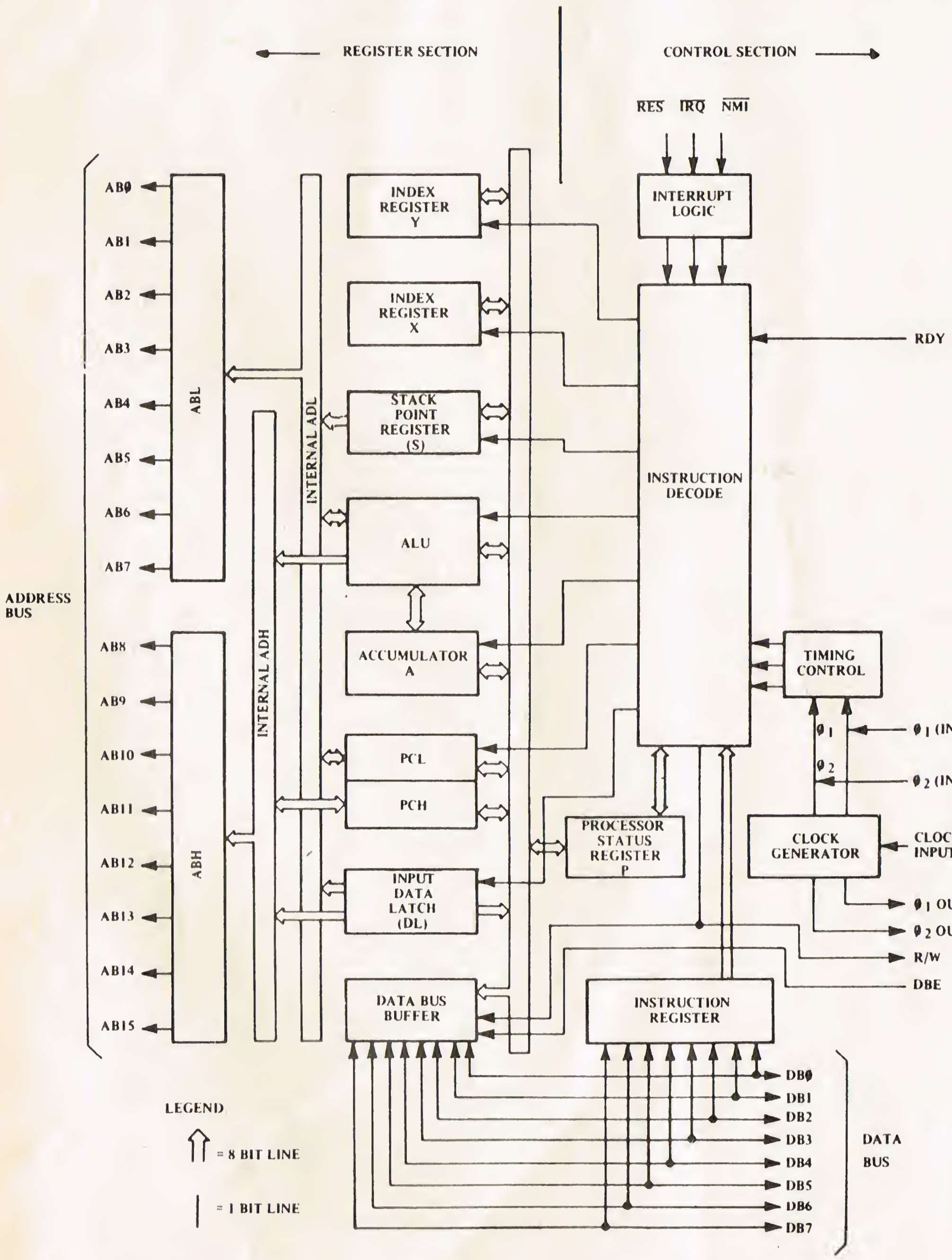
The 6500 series of processors are in many respects very similar to the 6800, but incorporate what MOS Technology feel are numerous improvements. The series is not as widely known as the 6800, its main exposure being through the KIM and PET.

The biggest and "best" of the line is the 6502, and to encourage the use of this mpu the 6501 was introduced. The 6501 is hardware compatible with the 6800, ie a plug in replacement, but incorporates the instruction set of the 6502 which is significantly different. The user is thus provided with a means to sample the 6500 series instructions (not compatible) without needing a complete system.

6501

As stated the 6501 is externally very similar to the 6800, the main difference being that VMA is not needed, since all addresses from the 6501 are valid.

The internal block diagram for the 6500 series processors is shown in figure 3. In contrast with the 6800, the 6501 contains only one 8 bit accumulator, two 8 bit index registers, a 16 bit program counter and 8 bit stack pointer. The status (condition code) register is again similar to the 6800 with the addition of one bit to set "decimal mode" of operation (BCD arithmetic) and one bit associated with software interrupts.



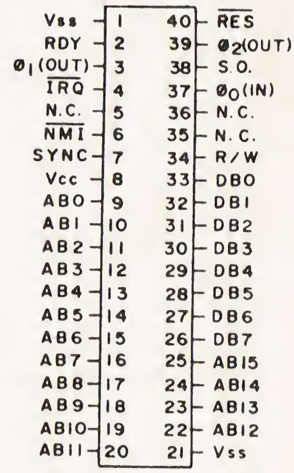


Fig 3. Block diagram of 6500 series chips. The pinout is for a 6502 mpu.

The switch from two to one accumulator might hamper some, but the change from one 16 bit index register to two eight bit ones is often advantageous. Index registers are generally used for table operations, where the base of the table is given with the op code, and the index register is added and incremented for each table access. Few tables are longer than 256 entries, but two tables might easily be used.

6500 software

A complete listing of instructions is given in figure 4 most of which are self explanatory. What is most interesting is the multitude of addressing modes. There are two major categories of addressing modes. The first of these, "direct" covers all the same address modes as those in the 6800. One small difference to the programmer does occur however. Where two bytes of address information follow the op code, in the 6800 system one writes them high order then low order. For the 6500 processors they are placed low order byte first, which at first looks odd but there is a good reason. Example: in a normal instruction execution, the address of the op-code is placed on the bus, and the op-code input to the instruction register. In the second cycle the op-code is being interpreted, while the next byte is loaded into an internal low order address register. Now if the interpretation of the op-code shows that it was a one byte instruction, the second byte was not necessary and is ignored. If a zero page instruction then the byte is already in the low address register, and if an absolute (two byte address needed) instruction another byte will be needed but at least the first one (low order) is already in the correct register. In the 6800 the op-code must be interpreted before the second byte can be put in the high or low register. This points out an important feature of the 6500 series, that every cycle is used for mpu input whether needed or note, (which is why no VMA is needed) but which often saves time.

The second category of addressing modes is the indirect category wherein the 6500s really shine.

This mode is a little complicated, so keep in mind that the objective is to get the operand. In plain indirect addressing the location following the op-code contains the address (in zero page) where two consecutive locations contain the address of the operand. Figure 5 shows the relationship. It is generally used where the final address (CB) is to be calculated at a later date, but the place to find that calculated address (A) can be established. To further beautify (complicate) matters, we can add indexing to either the second address (indirect indexing) or first address (indexed indirect). In the first case we may be accessing a table at a remote location, in the second perhaps accessing a table of remote locations, either way the final address is in some way to be left open for the program to calculate and insert. These modes are also most useful in subroutines.

6502

The 6502 includes a number of features over the 6501 and of course the same software features.

Fig. 4. The instruction set of the 6500 series mpus.

ADC	Add Memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift Left One Bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
BMI	Branch on Result Minus
BNE	Branch on Result not Zero
BPL	Branch on Result Plus
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set
CLC	Clear Carry Flag
CLD	Clear Decimal Mode
CLI	Clear Interrupt Disable Bit
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index X
CPY	Compare Memory and Index Y
DEC	Decrement Memory by One
DEX	Decrement Index X by One
DEY	Decrement Index Y by One
EOR	"Exclusive-Or" Memory with Accumulator
INC	Increment Memory by One
INX	Increment Index X by One
INY	Increment Index Y by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index X with Memory
LDY	Load Index Y with Memory
LSR	Shift Right One Bit (Memory or Accumulator)
NOP	No Operation
ORA	"OR" Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)
RTI	Return from Interrupt
RTS	Return from Subroutine
SBC	Subtract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode
SEI	Set Interrupt Disable Status
STA	Store Accumulator in Memory
STX	Store Index X in Memory
STY	Store Index Y in Memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index X to Accumulator
TXS	Transfer Index X to Stack Pointer
TYA	Transfer Index Y to Accumulator

Hardware

The block diagram figure 3 again applies, and the pinouts are shown in figure 3. The improvements are significant. A simple crystal or RC TTL single phase clock input is all that is required, the chip provides phase one and two outputs. A very important feature to hobbyists and prototype builders is the function of the RDY line. Similar to the HALT of the 6800, the difference is that the 6502 will stop with the addresses available on the bus, rather than in high impedance state. The advantage is that in debugging and single stepping it is very simple to stop and see where you are, and what data is at the location in memory. To do the same thing with a 6800 needs extra logic and latches.

Others in the 6500 series:

In addition several other processors are available, all "subsets" of the 6502. The 6503, 4, 5, 6 are 28 pin versions, with reduced addressing, control and interrupt combinations, on chip clock, and reduced cost. The 6512 is a 40 pin model with two phase clock but otherwise like the 6502. The 6513, 14, 15 are similar to the 6512 and again are 28 pin versions. 1MHz and 2MHz versions are available of each one.

Support chips

Since the buses for the two families are so similar, support ICs for one will generally work for the other.

The popular system staples are:

6810: 128 × 8 static RAM

6820: Peripheral Interface Adapter: provides two 8 bit parallel I/O ports, each bit programmable as in or out plus "handshaking". The two ports provide different input and drive capabilities, giving TTL and CMOS compatibility.

6830: 1024 × 8 ROM

6850: Asynchronous Communications Interface Adapter: Provides buffering and control for reception and transmission of serial data, eight and nine bit, with various code options.

6852: Synchronous Serial Data Adapter.

6860: Digital Modem for use with the ACIA, this unit provides the modulator, demodulator, and control signals for telephone communication.

6520: Peripheral Interface Device: similar to 6820.

6530: Peripheral Interface/Memory Device: includes 1K ROM, 64 bytes RAM plus two 8 bit bidirectional ports, and programmable interval timer with interrupt.

6522: Similar to 6520, plus timers, serial-parallel, parallel serial shift register, and input data latching, plus expanded handshaking.

6532: Includes 128 byte RAM, two 8 bit bidirectional ports and timer.

Plus an assortment of special purpose interface adapters and memory items.

CT

Program Location	Contents	Location	Contents	Location	Contents
X	Op-code	00,A	B	C,B	Data (operand)
X+1	A	00,A+1	C		

Fig 5. Indirect addressing on the 6500 series mpus.

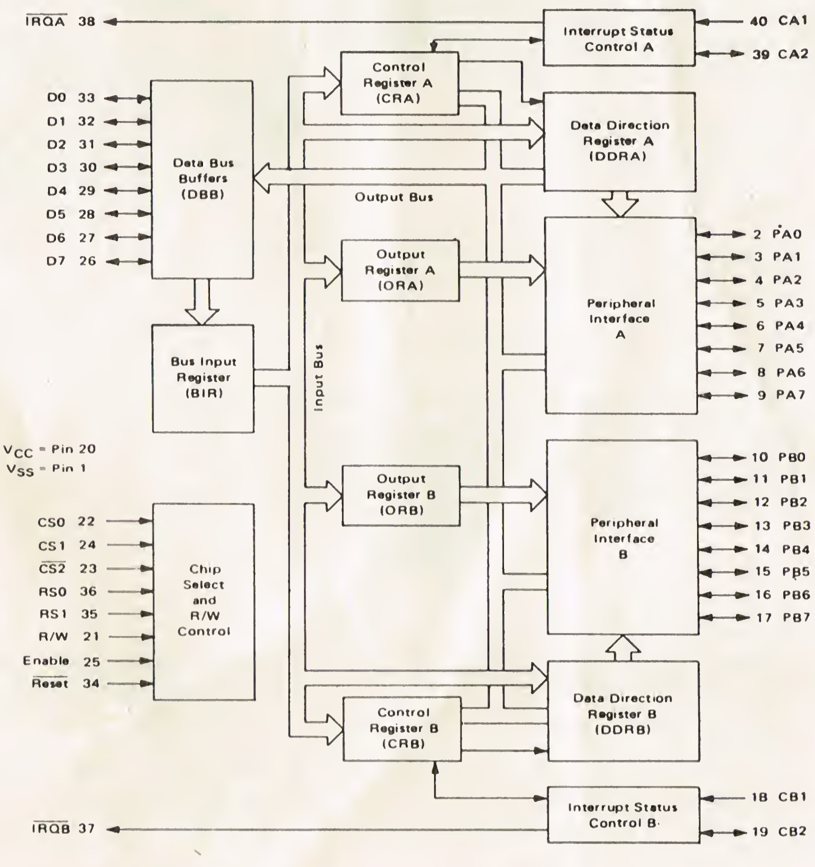
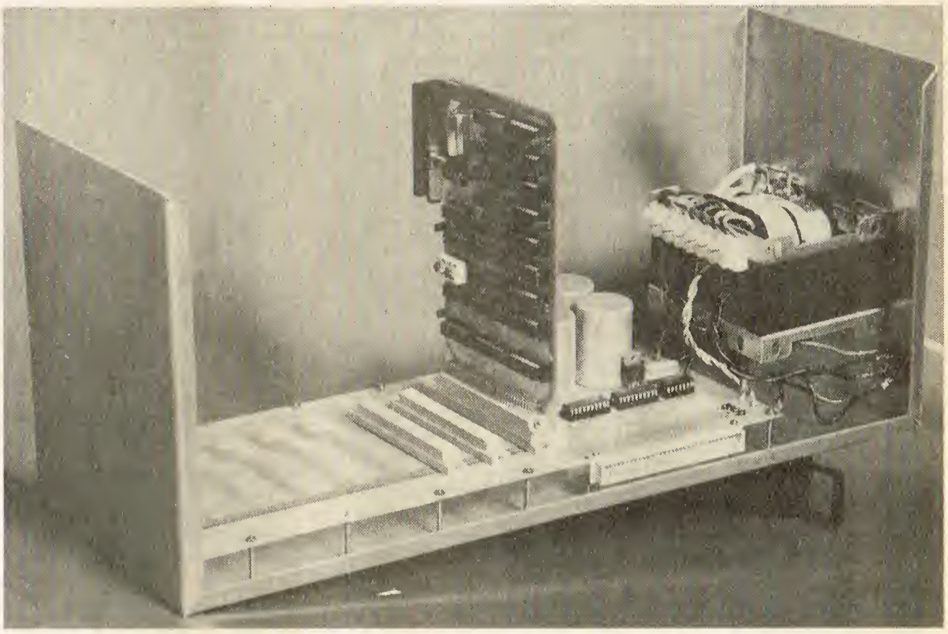


Fig 6. The Peripheral Interface Adaptor provides two 8-bit bidirectional points.

TRITON

Designed by Mike Hughes, complete kits will be available from Transam



Motherboard

Since the publication of our TRITON ONE BOARD COMPUTER in the November 1978 issue of ETI hundreds of readers have successfully completed the project and we have been inundated with requests for hardware extensions which will allow system expansion for more ambitious applications. We are pleased to be able to publish the first essential peripheral which will allow almost unlimited extensions. This is the MOTHERBOARD. Elsewhere in this issue we give the design of our first, and perhaps most important, peripheral board — an 8K static RAM card. This will shortly be followed by other peripherals all of which will be designed to plug into the motherboard without any changes to internal wiring.

YOUR MOTHER WOULD LIKE IT

In some respects the motherboard is a fairly uninteresting project in itself but it is a means to an end in that it opens the door for a whole host of possible additions to the TRITON. Although we shall be publishing what we feel to be the most generally useful additions individual users will be able to use the motherboard as an interface for any more specialised additions of their own design.

In order to keep the cost within bounds and to avoid too much in the way of redundant circuitry for the bulk of our readers we have designed the motherboard in such a way that when it is in use it will no longer be possible to carry out DMA operations on the TRITON system as a whole. This should only adversely affect a small number of readers but we feel sure that they would be able to embody the necessary modifications and additional circuitry to re-enable this feature if they particularly need it.

When you have read the accompanying article you will note that we have adopted a busbar pinning configuration which we shall now consider to be standard for the TRITON. Any further extensions that we pub lish will be based on this specification. If any readers eventually design interesting peripherals of their own we would be delighted to publish these for the benefit of others. We beg you, however, to keep any such designs to a single EUROCARD format (custom built or Veroboard pcbs can be used) and adhere strictly to the busbar format. Ideally the designs should allow flexibility in the selection of board or port addresses and provide a correctly formatted $\overline{\text{DINE}}$ signal (see how the motherboard works for details of this).

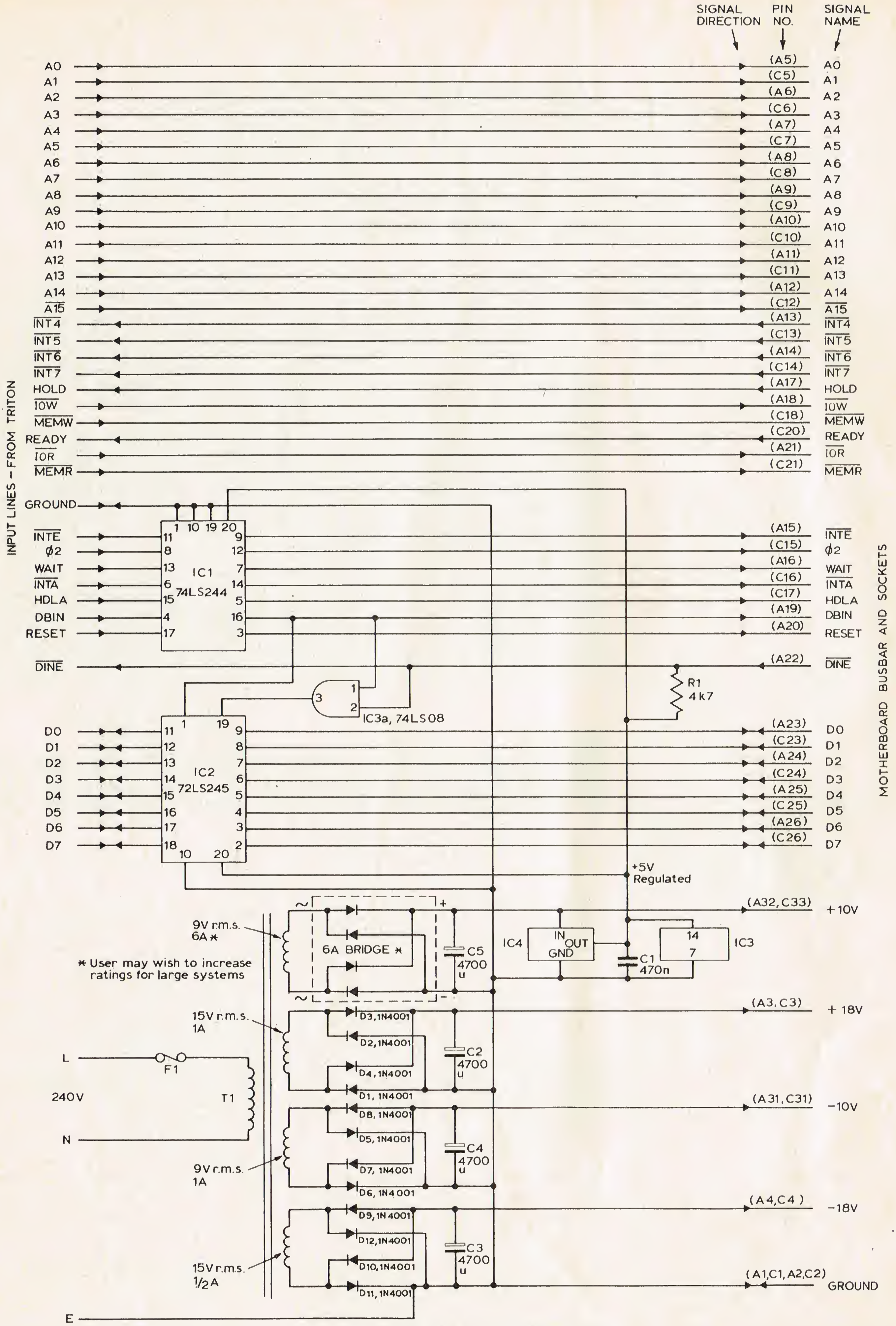
TAKE CARE OF MOTHER

The motherboard is a high quality double sided board with plated through holes with all closely packed busbar wiring on the topside — well away from the soldering iron. Because of this it is extremely simple and easy to assemble. Space is provided for eight peripheral boards plus an output socket — which will allow one motherboard to be cascaded into another should the need ever arise. As the 64 pin sockets are quite expensive most readers will not wish to insert all of them in the first instance but the bare minimum will be the input socket and a further socket for one peripheral board. As we already have a design for an 8K Static RAM card and are well advanced with the design of an 8K EROM extender board it might be worth considering the insertion of three peripheral board sockets but this is entirely up to the individual.

Start assembly by inserting **all** board pins (including all those required for the I/O lands). It is also sensible to drill out all fixing holes to suite your screws but do not, under any circumstances, attempt to drill out electrical connection holes.

Insert all the electronic components according to the overlay drawing taking great care that you observe the correct polarity of the electrolytic capacitors. You should then insert, and bolt into place the board's input socket. **Note** that this is inserted from the **underside** of the board unlike all the other sockets. **It is crucial** that this is inserted from beneath otherwise the whole of the busbar integrity will be reversed with possibly disastrous results. If you get this wrong you will, at best, ruin the PCB when you try to remove the 64 pins after they have been soldered!

To finish off the board you should insert and solder in as many board ⋙



SCHEMATIC OF TRITON MOTHERBOARD

TRITON OUTPUT SOCKET		MOTHERBOARD INPUT SOCKET	
SIGNAL	PIN	PIN	SIGNAL
READY	C1	C20	READY
RESET	C2	A20	RESET
ϕ2	C3	C15	ϕ2
$\overline{A15}$	C4	C12	$\overline{A15}$
A11	C5	C10	A11
A14	C6	A12	A14
A10	C7	A10	A10
A13	C8	C11	A13
A12	C9	A11	A12
$\overline{INTA}$	C10	C16	$\overline{INTA}$
$\overline{INTE}$	C11	A15	$\overline{INTE}$
WAIT	C12	A16	WAIT
$\overline{IOW}$	C13	A18	$\overline{IOW}$
$\overline{IOR}$	C14	A21	$\overline{IOR}$
HOLD	C15	A17	HOLD
$\overline{INT4}$	C16	A13	$\overline{INT4}$
$\overline{INT5}$	C17	C13	$\overline{INT5}$
$\overline{INT6}$	C18	A14	$\overline{INT6}$
$\overline{INT7}$	C19	C14	$\overline{INT7}$
D4	C20	C24	D4
D3	C21	A24	D3
D2	C22	C23	D2
D1	C23	A23	D1
A9	C24	C9	A9
A8	C25	A9	A8
$\overline{MEMW}$	C26	C18	$\overline{MEMW}$
$\overline{MEMR}$	C27	C21	$\overline{MEMR}$
D5	C28	A25	D5
D6	C29	C25	D6
D7	C30	A26	D7
D8	C31	C26	D8
GROUND	C32	A1	GROUND
A7	A23	C8	A7
A6	A24	A8	A6
A5	A25	C7	A5
A0	A26	A5	A0
A1	A25	C5	A1
A2	A28	A6	A2
A4	A29	A7	A4
A3	A30	C6	A3
DBIN	A31	A19	DBIN
HDLA	A32	C17	HDLA

WIRING DIAGRAM OF TRITON / MOTHERBOARD COUPLING CABLE

NOTE
Once prepared this cable is NOT reversable so the respective sockets should be clearly marked TRITON and MOTHERBOARD.

Keep cable length as short as possible, preferably not more than 75cm.

sockets as are necessary. Note that the socket lands at the extreme end of the board are for a right angled fixing socket for use as an output connector. **Take special care** to ensure you insert the board sockets the right way round so that the pin numbers conform with those shown on the overlay drawing. Excess length of the wire wrap pins can be clipped off when soldering is complete.

It only remains to mount the board within the metal chassis using suitable spacers (ensure that the height of the input socket will match the aperture of the cabinet's cover) and complete the wiring with the transformer and bridge rectifier.

Testing time

Now comes the more tedious job of wiring up the interconnection cable which links the TRITON main board with the motherboard. Do not make this cable too long (preferably keep it less than 75cms). We used two sockets-identical to the type used as board sockets on the motherboard — but cut the wire wrap pins to about 5mms length to avoid the possibility of shorts. Remember to slide the socket covers on to your wiring loom **before** you solder up both ends of the cable! We forgot and lived to regret it when we had to unsolder and resolder 42 connections!

Take great care in making up this cable to ensure that you follow the pin to pin wiring chart precisely and when you have completed it you should »»

How It Works

The motherboard's function is to permit extensions of memory or input/output peripherals for the TRITON main board. These additions can be built on single Eurocards and plugged into one of the eight socketed locations on the motherboard. Each card which is added can draw power from a common unregulated busbar which is fed to specified pins on each socket. These power rails carry raw d.c. of +18V (1A), −18V(½A), +10V (6A) and −10V(1A). These voltages have been chosen to give standard ±12 and ±5V after regulation-it being assumed that most current will be needed at +5V and the least at −12V. Whether or not these currents are sufficient will depend very much on the types of board plugged in but it is assumed that there will be sufficient power availability for most applications. If, for very large systems, more current is required (typically at +5V) it is only necessary to upgrade the transformer and the respective rectifiers. This will probably not be necessary but anyone who intends designing his own extensions should remember to take note of current consumptions.

Apart from providing power to its sockets the motherboard also contains a minimal number of integrated circuits. IC1 contains buffer for the seven unbuffered outputs from the TRITON main board and IC2 is a bi-directional buffer (with tristate outputs) to interface the data busbar with the motherboard. A small amount of logic is needed to control the enabling of this data buffer when it is transmitting towards the TRITON main board. This logic is provided by a simple AND gate. The $\overline{\text{DBIN}}$ (Data Bus In) signal controls the direction of the buffer and when this is "0" (i.e. indicating an output from the CPU) the motherboard's data bus is enabled irrespective of any other conditions. When DBIN is "1" the buffer is **not** enabled unless the special $\overline{\text{DINE}}$ line is at "0". The DINE signal (Data In Enable) is obtained by wired oring of a suitable signal from each of the peripheral boards. This signal should be obtained from the decoded board select address in the case of memory boards but in the case of Input/Output port addressing it should be obtained by ANDING the respective Board/ Port Select with $\overline{\text{IOR}}$. The gate supplying this signal from each board **must** be of the open collector type.

Apart from the foregoing the motherboard acts as an interconnection between the TRITON main board and the peripheral boards by carrying the complete address and control busbars which are adequately buffered within TRITON itself.

The respective signals on TRITON's main board extender socket are not organised in any particular order and it is necessary to feed these to the input socket of the motherboard via a 42 way cable. Note that $\overline{\text{DINE}}$ does **not** connect back to the TRITON main board. It is most important to adhere precisely to the pin numbering configurations for this cable the details of which are shown in Fig 0. We do not claim that the TRITON BUS conforms to any other standard but the pin configurations for power, address and data lines are the same as those specified for Eurobus.

Each peripheral board socket has eight spare connections brought out to soldering lands on the edge of the motherboard. These are to provide Input or Output lines from specific boards which can be wired to output sockets on the end panel of the cabinet. These are user definable but we shall be making use of some of these in future projects involving peripherals. Readers will note that there are two spare Bus lines running through the motherboard — these are on pins C19 and C22 — they may be used for special applications or the bus track can be broken and the pins used to provide an extra couple of I/O outlets.

Parts List

RESISTORS
R1 4k1 10% ¼W

CAPACITORS
C1 470n polyester
C2 4 700uF 25V
C3 4 700uF ,,
C4 4 700uF ,,
C5 4 700uF ,,

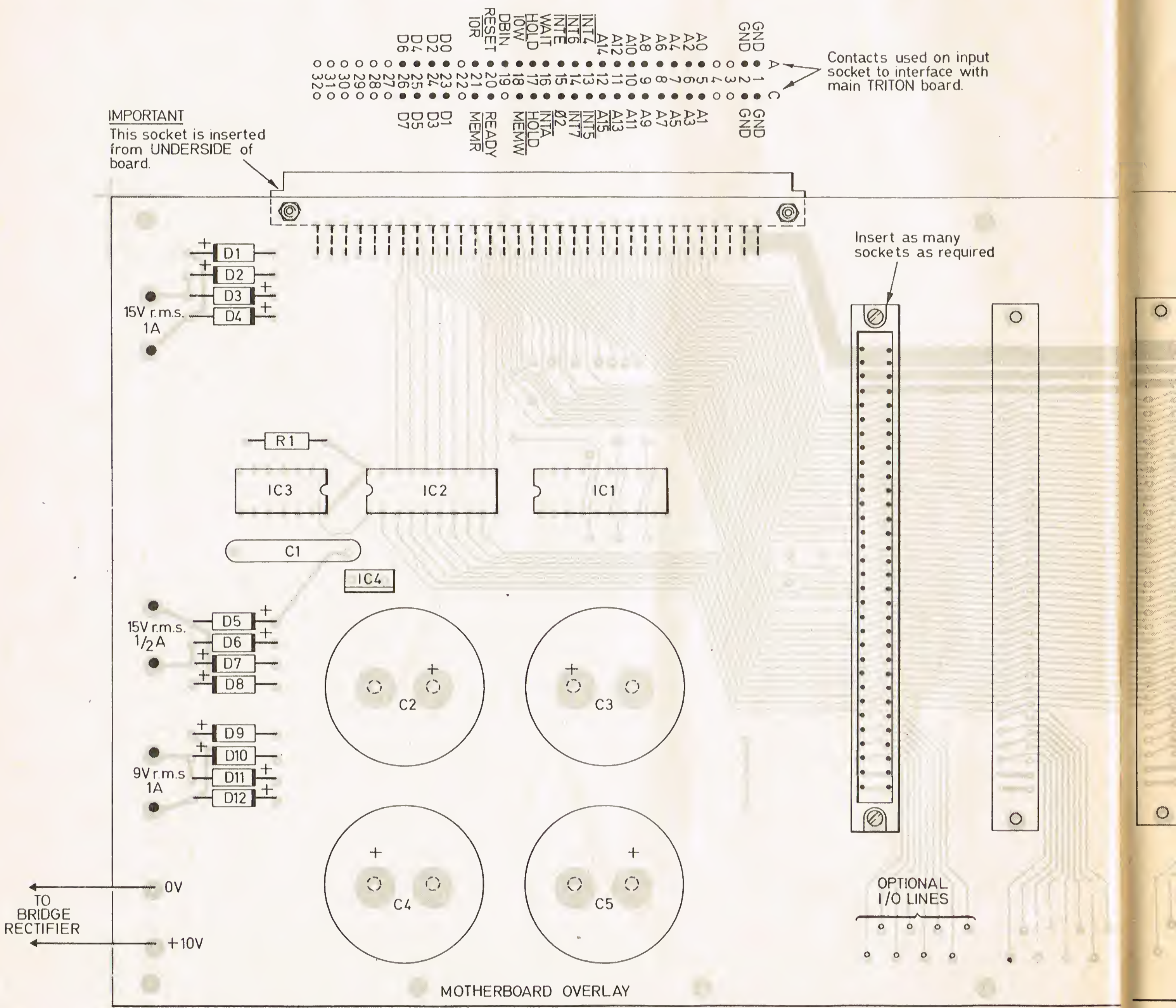
SEMICONDUCTORS
D1 to D12 1N4001 (25V 1A rectifiers)
BRIDGE 25V 6A
IC1 74LS244
IC2 74LS245
IC3 74LS08
IC4 7805C +5V plastic regulator

MISCELLANEOUS
T1 Mains transformer 15V, 1A; 15V, ½A; 9V, 1A; 9V, 6A secondaries.
F1 panel fuse holder (1A)
Printed Circuit Board
Input socket (Plus optional output socket)
1 (or more) board sockets
2 interconnecting cable sockets and covers
Metal chassis and cover
2 off 20 pin DIL sockets
1 off 14 pin DIL socket

permanently mark each socket to show which end of the cable is which — **it is not reversible.**

Our suggested test of the motherboard is rather negative because — on its own — it does not do anything. None the less it is worth doing the following to ensure that nothing particularly wrong has happened.

With TRITON switched **off** plug in the motherboard to TRITON making sure you have used the cable the right way round and apply power to the motherboard. Check that you have +5V at the respective pins of the three



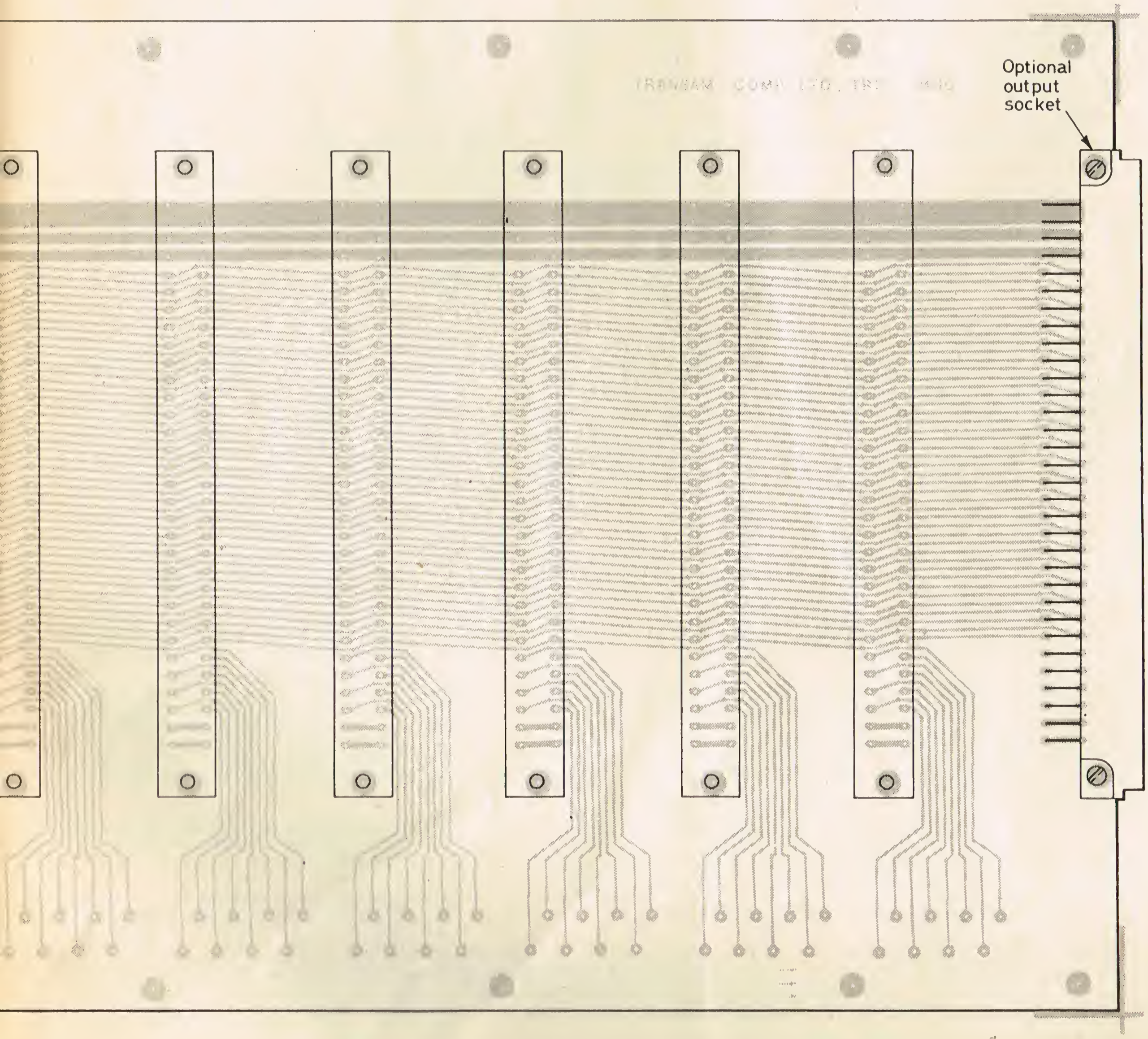
MOTHERBOARD OVERLAY

integrated circuits. Check also that you have approximately the correct nominal voltages on the four power busbars. Because there is no load at present these might read a little high. You should see around ±19 and ±11 volts.

When you are satisfied that all this is well you should switch on the TRITON and it should initialise as if nothing had happened. If you get normal initialisation followed by the repeating message "INVALID" check that you have put in the modifying link taking the +5V rail straight to pin 28 of IC6 on the TRITON main board (we have drawn attention to this elsewhere in the magazine).

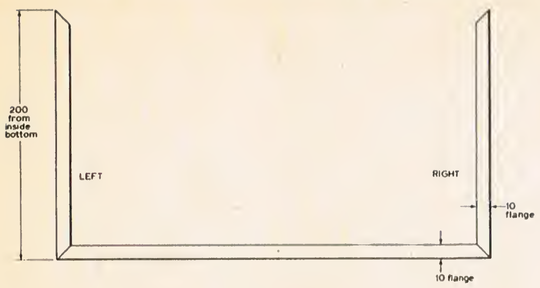
If your TRITON has always operated satisfactorily until now but refuses to initialise you can be pretty sure that you have a short circuit between tracks of the data, address, or control busbars on, or feeding to, the motherboard. If this is the case you should switch off and disconnect the linking cable and systematically check **every** possible combination of the 42 wires for continuity (going from the right pin of one plug to the right pin of the other) and shorts between adjacent pins. Following on from this you should carefully check every combination of bus line pairs on **both** sides of the mother board buffers.

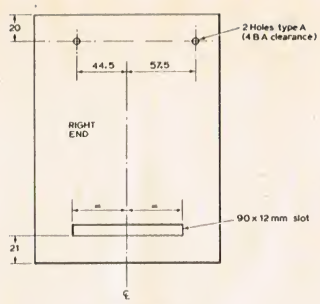
When all is well and your computer initialises you can be pretty certain that the motherboard is working correctly but you cannot know this for sure until you have a peripheral board plugged into it. When using TRITON with the motherboard you must always make sure that power is applied to the motherboard **before** you switch on the TRITON.



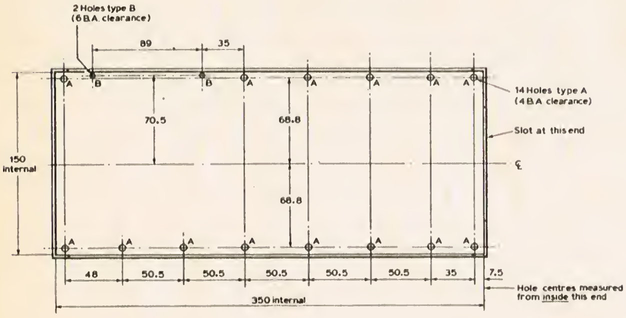
	A		C	
GND	•	1	•	GND
GND	•	2	•	GND
+18V	•	3	•	+18V
−18V	•	4	•	−18V
A0	•	5	•	A1
A2	•	6	•	A3
A4	•	7	•	A5
A6	•	8	•	A7
A8	•	9	•	A9
A10	•	10	•	A11
A12	•	11	•	A13
A14	•	12	•	A15
INT4	•	13	•	INT5
INT6	•	14	•	INT7
INTE	•	15	•	Ø2
WAIT	•	16	•	INTA
HOLD	•	17	•	HDLA
IOW	•	18	•	MEMW
DBIN	•	19	o	*
RESET	•	20	•	READY
IOR	•	21	•	MEMR
DINE	•	22	•	*
D0	•	23	•	D1
D2	•	24	•	D3
D4	•	25	•	D5
D6	•	26	•	D7
	o	27	o	
	o	28	o	
	o	29	o	
	o	30	o	
−10V	•	31	•	−10V
+10V	•		•	+10V

CONTACTS USED
ON OUTPUT SOCKET
* DENOTES SPARE BUS LINE

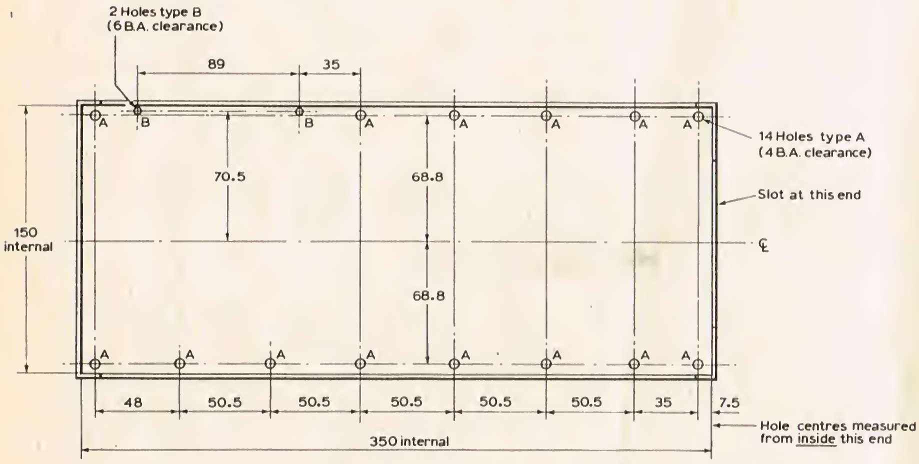


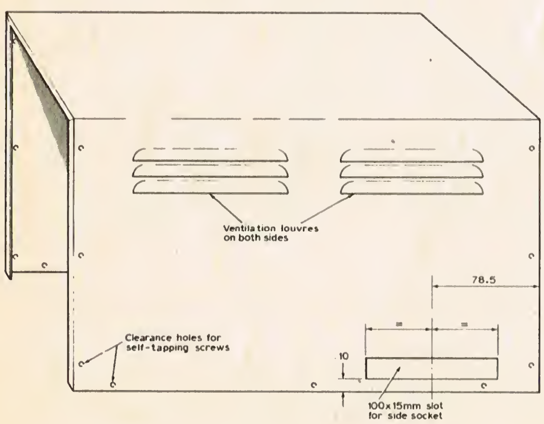


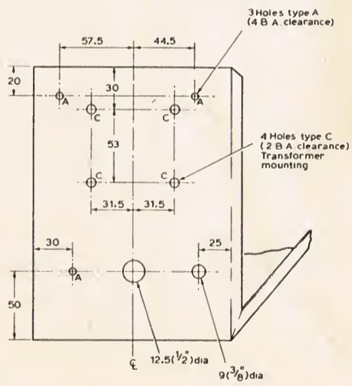
THIRD ANGLE PROJECTION
all dimensions in mm



MOTHERBOARD CHASSIS
Fabricate from 2mm aluminium Supply complete with 4 rubber feet and cover (attached with self-tapping screws).
See second drawing for details of cover and LEFT END drilling







DRILLING DETAILS OF LEFT END MOTHERBOARD CHASSIS
All dimensions are in mm.

RAM Card

REMEMBER, REMEMBER

This 8K static RAM card contains 8192 contiguous bytes of read write memory organised in blocks of 1K. For reasons of economy some people might wish to expand their system in easy stages in which case the chips can be added to this board in units of 1K. The board itself can have its start point address selected by means of a jumper wire (or DIL switch) and can be positioned at the start of any 8K region in a 64K memory map. In the case of TRITON the lowest order 8K region is entirely taken up by the main board so it is assumed that the first RAM extension card should be positioned to start at 2000H to make it contiguous with existing memory.

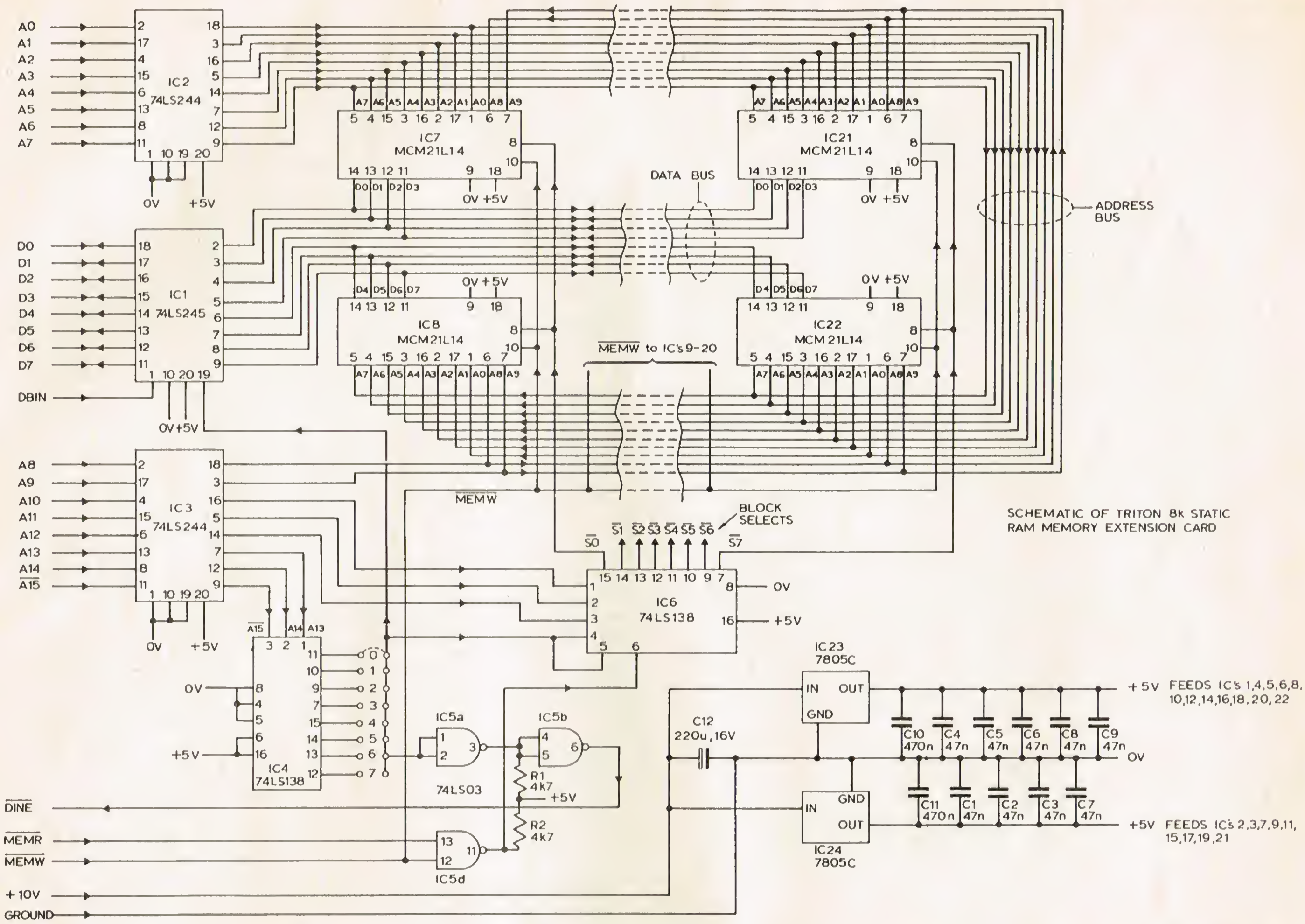
The card is designed to plug directly into the TRITON MOTHERBOARD and draws, typically, 1A from the +10V unregulated power bus. It is designed specifically to interface with the TRITON BUS configuration and provides the correct $\overline{\text{DINE}}$ signal; no claim is made that it will operate in other systems without modification.

It is built on a specially designed double sided Euro card and assembly is extremely straightforward — average building time being about 3 hours.

Insert and solder all integrated circuit sockets **one at a time** ensuring that no connections are missed. Start with the larger sockets — this will prevent you accidentally soldering in an 18 pin socket where there ought to be one having 20 pins etc! We suggest using a 16 pin DIL socket to carry the board select jumper wire so that this can be altered to reposition the board's address should the need arise. Where expense is no object and for development systems you could, as we did, use plug in DIL switches in this socket.

When all sockets are in position make sure you solder in the two pull up resistors.

The next stage is to bolt and solder in the 64 way Eurosocket. This is inserted from the component side of the board. Move on to the two +5V regulators making sure you insert them the right way round and then ›››



SCHEMATIC OF TRITON 8k STATIC RAM MEMORY EXTENSION CARD

How It Works

IC1 is a bidirectional buffer for the 8 bit data busbar and receives and transmits data to the motherboard. Its direction of operation is controlled by the DBIN signal but its outputs are not enabled unless the board recognises its own address (decoded by IC4) — this enabling signal is a "0" on pin 19 of IC1. The 16 bits of the address busbar are buffered by ICs 2 and 3 which are permanently enabled.

The three highest order bits of the address busbar (A15, A14 and A13) are used to decode which 8K region the board sits in. There are 8 possibilities so we are using a three to eight line decoder. Only one of the eight possible outputs will go to "0" for a given high order address and we can select one of these outputs to identify the board's starting address. This is done by means of a wire link. We anticipate that all our memory extension cards will be of 8K capacity therefore it is convenient to think of the 8 possible regions of 8K being designated 0, 1, 2 etc. up to 7. The memory map of the TRITON system can then be simply defined by the following chart:

Region	*Address limits*	*Description*
0	0000H—1FFFH	TRITON main board embracing EROM, VDU and RAM
1	2000H—3FFFh	User definable
2	4000H—5FFFH	User definable
3	6000H—7FFFH	User definable
4	8000H—9FFFH	User definable
5	A000H—BFFFH	User definable
6	C000H—DFFFH	User definable
7	E000H—FFFFH	User definable

Region 0 is already spoken for by the TRITON main board so none of the extension cards must sit in this area. We would expect that most people will wish to have continguous memory (i.e. an unbroken sequence of addresses) starting from 1600H which is the start of work area on the main board so the first RAM extension card should be situated in Region 1, the next in region 2 etc. Any specialised memory could then be sited at the high order end of the map (regions 6 and 7). These comments are, of course, generalisations and individual users can arrange their memory in any order they wish using the 8K regions as building bricks.

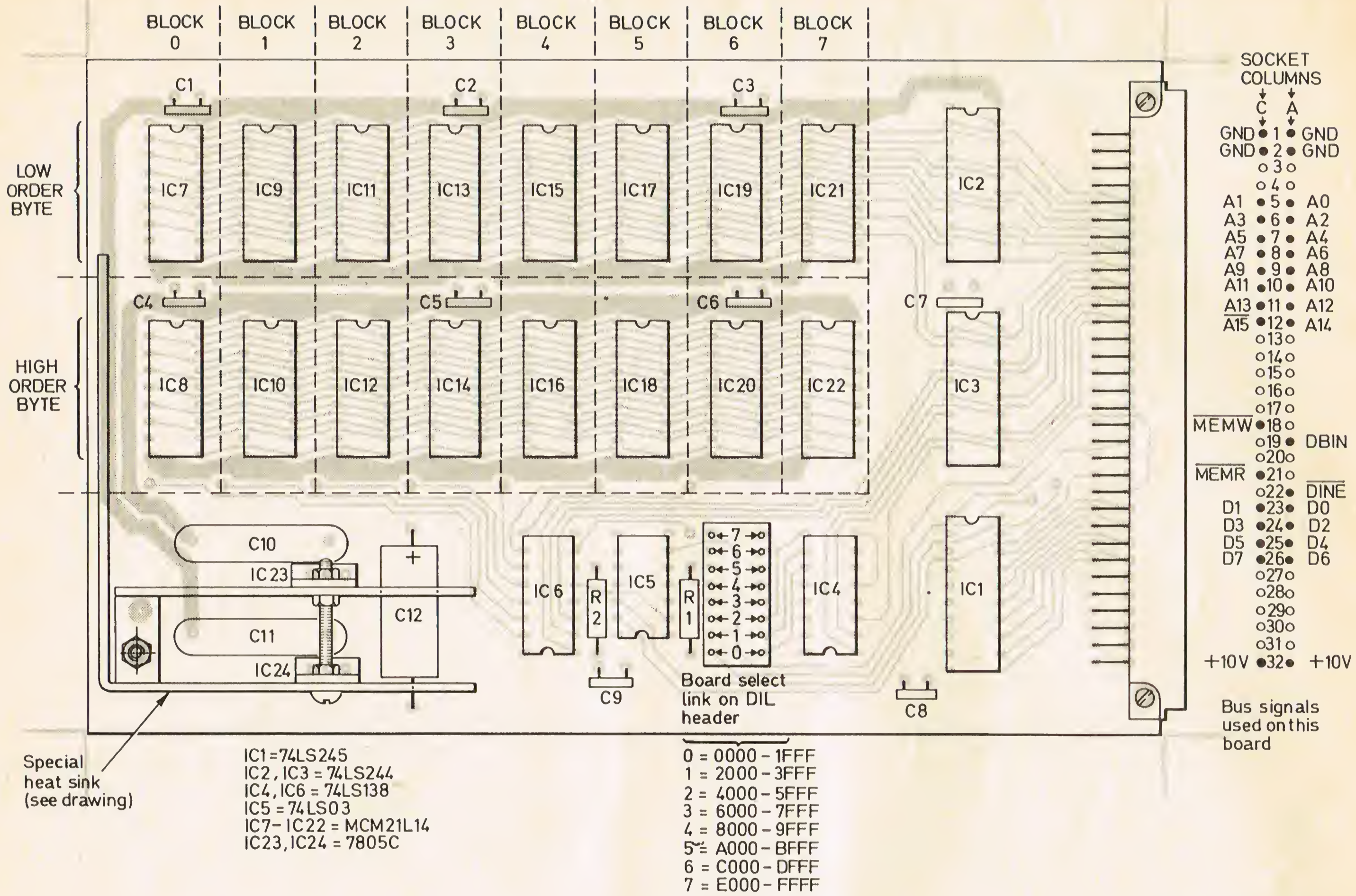
In the case of this static RAM card we are using pairs of 1K × 4 memory chips type MCM21L14 (note we are specifying the low power version to economise on total current consumption — the higher power versions CAN be used but board dissipation might get uncomfortably near the limit of the voltage regulators). This allows us to organise the memory within the board into 8 blocks of 1K apiece. These can be conveniently and individually selected by the three next lower address lines (A12, A11 and A10) through a further three to eight line decoder (IC6). We have to use the board select signal to enable this decoder so that the blocks are selected ONLY when the board as a whole is addressed. This enabling signal is fed into pins 4 and 5 of IC6. We have designated the eight block selects signals (active low) as S0 to S7. S0 represents the lowest order block of the board so for contiguous memory you should insert memory chips into locations IC7 and 8 in the first instance and then works upwards (that is if you do not intend to put a complete load of RAM into the board in the first instance).

As the RAM chips are organised as 4 bit wide "nibbles" we have to use two Integrated circuits for each block. Odd number component designations (IC7,9 11 etc) represent the low order nibbles and even numbers (IC8, 10 12 etc) the high order.

The ten lowest order address lines are parallelled to all the RAM chips on the board in the usual way but please note that we are NOT using the data sheet's designations for the addresses. The reason for this is tied into the convenience of the board's layout. It makes no difference to the operation but some people might query why we have done this. A small amount of logic is needed to organise the MEMR and MEMW command signals to interface these with the RAM's writing and select inputs. This is provided for by the NAND gate (IC5d). The logic ensures that a select line becomes active whenever either the MEMR or MEMW lines go low but the WRITE input of the RAM is only taken low when MEMW is active.

ICs 5a and 5b take the board select signal and output this as an active low through the open collector gate (IC5b). This is the DINE (Data In Enable) which can be wire ored with similar signals from other boards on to the motherboard busbar. This is used to enable the motherboard's data bus buffer during memory or I/O read operations.

Two 1A regulators are used to provide +5V.



8k STATIC RAM CARD OVERLAY

insert all the capacitors checking the polarity of C12.

Assemble the special heat sink to the regulators making sure that the fixing bolts are really tight — quite a lot of heat is dissipated and the heat sink size is a bare minimum. If you make your own from our drawing we recommend that you paint it black to increase its radiating efficiency. Check that no parts of the heatsink touch any components (particularly the three capacitors in close proximity) or any of the topside foil tracks (one of the +5V rails goes fairly close to the board mounting hole of the heatsink). Insert all integrated circuits with the RAM starting at block "0".

To test the board make up a DIL pin header with a jumper wire bridging the "1" position (this positions the

Further thoughts on Triton

From talks with TRITON users and feedback from the hundreds of people who have their systems up and running we are now able to pin point one or two features which might help those who are just about to start the project.

1) By far the greatest cause of systems not working in the first instance is that integrated circuits have not been correctly inserted into their sockets. The most common fault is in having a pin bent under (this happens during insertion and is difficult to notice unless the ICs are systematically removed, checked and replaced).
2) The second cause is the missing of one or two soldered connections. We repeat, again, the suggestion that you solder in ONE socket at a time and check it thoroughly BEFORE going on to the next.
3) If the system works satisfactorily with the Monitor plugged in but fails as soon as the two BASIC chips are inserted this is probably due to the design error we have already referred to. You should make sure that R1 is a 39 ohm 1½W resistor and ZD1 is up-rated to 5V1 at 1W.
4) From experience we have noticed that one or two systems suffer from a power line "glitch" which affects the system control chip (IC6). The symptom of this is when the computer occasionally prints a "-" of its own accord and, when under the control of the Monitor in its initialisation condition, follows this with the statement "INVALID". This problem can be easily cured with an insulated wire link (as short as possible) from the top side track feeding pin 28 of IC6 to the wide +5V track (on the topside) adjacent to R1. Even if you do not, at present, suffer from this problem you would be well advised to put this link in.
5) We recommend that you do NOT use the board mounting hole situated next to RV1 because it is rather too close to one of the underside tracks of the data busbar. If you use a nut instead of a stand-off you might short this track to ground. The symptom of this is that the system initialises but instead of printing the normal message a weird selection of letters and graphics is presented.
6) If you have trouble with the Tape I/O and are using a

board to start at address 2000H). Plug the board into any slot of the motherboard and connect the motherboard to TRITON.

Test with Triton

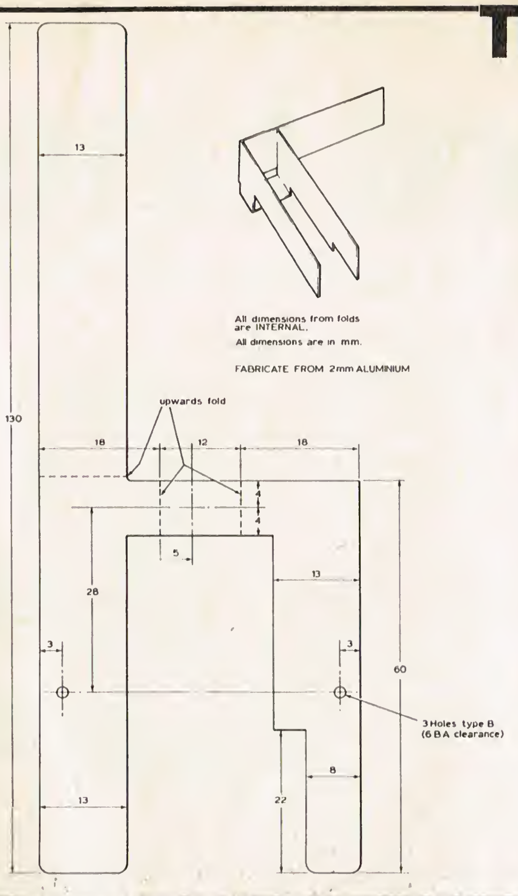
Do not apply power to TRITON at this stage but switch on the motherboard's supply. Check that +5V exists on both main rails of the RAM card and wait a few minutes for the heatsink to get up to temperature. It will run fairly hot — up to about 50°C.

We will assume that you have completely filled the 8 blocks of RAM of the card for the following tests. While the motherboard is switched on you should now apply power to TRITON. (NOTE always switch the motherboard on BEFORE you switch on TRITON). You should notice a slightly longer delay before the VDU screen is cleared and you get the normal initialisation message.

Using the P function check the contents of address locations 1481 and 1482. If you have a full main board and have inserted all 16 RAM chips on the RAM card the data contained in these two locations should be 00 and 40 respectively. This indicates that the first address where there is no RAM starts at 4000H. If any other data is found in these positions it will tell you the address at which your memory has failed and this should be of help in finding the fault (probably an IC incorrectly inserted).

If all is well you can try writing and reading within the memory region 2000H and 3FFFH.

If you make up more than one RAM board you must make sure that you use different address selections for them or else very strange things will happen! **CT**



Parts List

RESISTORS
R1 4k7 10% ¼W
R2 4k7 " "

CAPACITORS
C1-C9 47n ceramic
C10-C11 470n polyester
C12 220u 16V electrolytic

SEMICONDUCTORS
IC1 74LS245
IC2 74LS244
IC3 74LS244
IC4 74LS138
IC5 74LS03
IC6 74LS138
IC7-IC22 MCM21L14 (or similar)
IC23-IC24 7805C (+5V 1A regulators)

MISCELLANEOUS
PCB
64 way Eurosocket
Heat sink for regulators (see drawing)
3 off 20 pin DIL sockets
16 off 18 pin DIL sockets
3 off 16 pin DIL sockets
1 off 14 pin DIL socket
16 pin DIL header (for wire link)

mains driven tape recorder it is quite likely that you are suffering from an "Earth loop" problem which introduces a high level of hum on the tape input line. This is simply cured by breaking the mains earth connection on the tape recorder — thus relying on the braid of your coaxial cable to bond the system together.

7) RLY1 is only a small reed relay and is not rated to switch more than 1A at LOW VOLTAGE. Excessive voltage and current could cause the contacts to weld together!

8) If, when operating in BASIC, you get "Sorry" every time you enter an instruction it means that the BASIC INTERPRETER thinks it hasn't any memory. This could be a fault associated with ICs 29 and 30 or, more likely, is because you have — previously — been playing around in machine code and have inadvertently altered the table area of RAM which tells BASIC how much memory there is. You can check this by looking at address locations 1481 and 1482 (Hex) these should contain the address of the first location where RAM ceases to exist — for a board full of RAM the data contained therein should be 00 and 20 respectively. You can either manually enter this data or switch the system off and on again to carry out a new "power on initialisation".

9) There is a very minor bug in the Monitor which means that when you initially jump to BASIC and immediately ask it to PRINT SIZE without there being any program present it MIGHT give an erroneous answer if address locations 1600 and 1601 do not contain any valid end of text address. The simplest way of avoiding this bug is to enter "NEW" before starting to write any program in BASIC. This problem does not occur if BASIC is used immediately after switch on, only if you have been doing other things in machine code before jumping to BASIC.

10) You can get a very much higher resolution display by using a video monitor instead of a conventional television set. A suitable signal can be extracted from the negative end of C23. Together with a common ground connection this should be fed to a 75 ohm terminated video monitor input. UNDER NO CIRUMSTANCES feed this signal to a modified domestic television set unless you are certain the television ground is properly isolated from mains!

Hard Lines

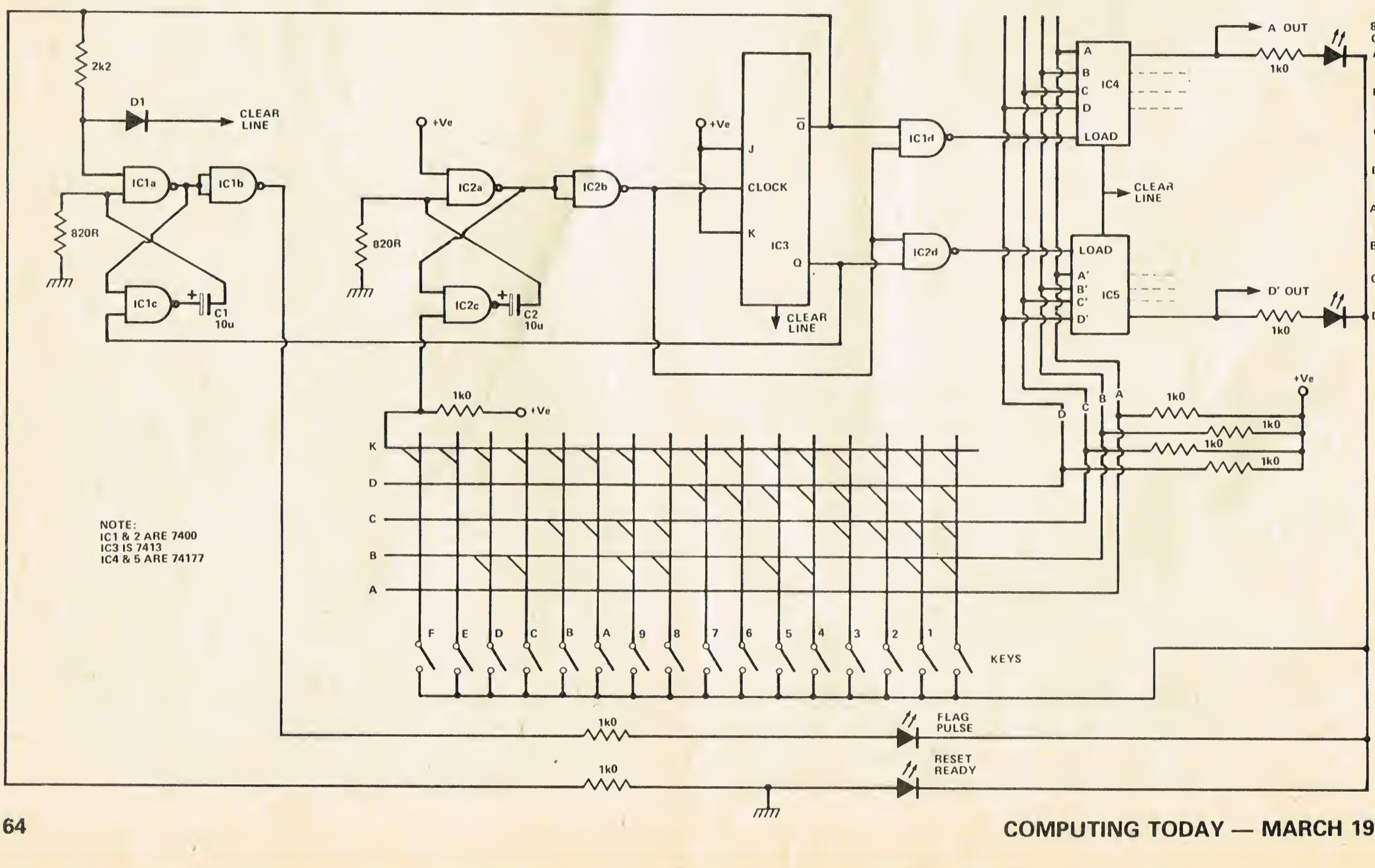
Richard Straker has sent us this hardware tip.

Having plucked up enough courage to go for a byte from the everswelling microprocessor apple it soon became apparent that there was more than one obstacle between me and the fruit tree. How to provide the eight bits in a row for the CPU without using DIL switches and probably losing some of the fingernails to which I have become so attached, was the first problem and the one that started this article.

IC4 and 5 are programmable 4 bit latches with active LOW load lines. Data on the four input lines transfers to the output lines when the load line goes low and is latched there when the load line goes high. Therefore if the load lines are made to go low in turn and the appropriate data is fed to the input lines, then an 8 bit byte is produced by combining the two 4 bit outputs. IC2, 3, and ¼IC1 in conjunction with the Key line from the keyboard cause the access of firstly IC4 and then IC5 and then back again to IC4 and so on. IC2a, b, c make up a monostable and inverter fed from a negative going pulse from the Key line. The monostable avoids keybounce problems. The positive going pulse from IC2b is fed to the joined gates of IC1d and IC2d and the clock of the JK flip flop IC3 which toggles. Thus in the reset/clear condition with Q=1, IC1d and hence IC4 are ready to receive a data load pulse. On the 1st key operation a pulse is produced which causes a low on the load line of IC4 and data is latched. On the downward slope of the pulse from IC2b, IC3 toggles to Q=1 and Q=0. Therefore IC2d and hence IC5 are now ready for their data load pulse. This occurs on the second key operation. However as IC3 toggles to its original reset condition Q goes low causing monostable IC1a, b, c, to produce a positive pulse which fires the flag line and the associated flag LED, indicating a full 8 bits worth of data. This flag can be used to drive the CPU equipment. These flag pulses are inhibited when the reset/clear is operated by the earthing of D1 which holds low the input on IC1a. This holds the output of IC1a at 1 despite the charging of C1.

The reset/ready LED is lit when IC3 is in the reset condition, that is before or after a full keyboard operation. It may be considered superfluous as an indication of data present is given by the 8 output line LEDs themselves. (Except at 0000 0000 of course).

The keyboard encoding is self-explanatory, the ABCD data lines being held high until taken low by the appropriate key to earth. Hence to encode 0 requires 4 diodes and F none. Each key requires a diode to Key line. Construction of the matrix, because the whole of the circuit was to be inserted into an old LED calculator case incorporating the keypad, had to be small. Therefore the diodes were sandwiched between two pieces of Vero board with tracks at right angles to each other, facing outwards. The diodes were soldered to the main board on 16 tracks first and then the other ends fed carefully through the smaller piece of Vero with 5 tracks. BEFORE the other ends are soldered positioning and continuity must be checked as replacement of diodes later would be next to impossible. This fiddly chore is worth it as the final construction is only about ·4 inch deep (including solder blobs). **CT**



Softspot

STOMPER is a program by Pete Howells which should appeal to the meaner side of us all. The object of the game is to chase an insect around the screen and kill it.

The program, as listed, is suitable for running in 4K on a PET. It is the result of sitting down at the machine and doing it, and, as you can see, in no way have the most efficient or elegant solutions to the problem been used.

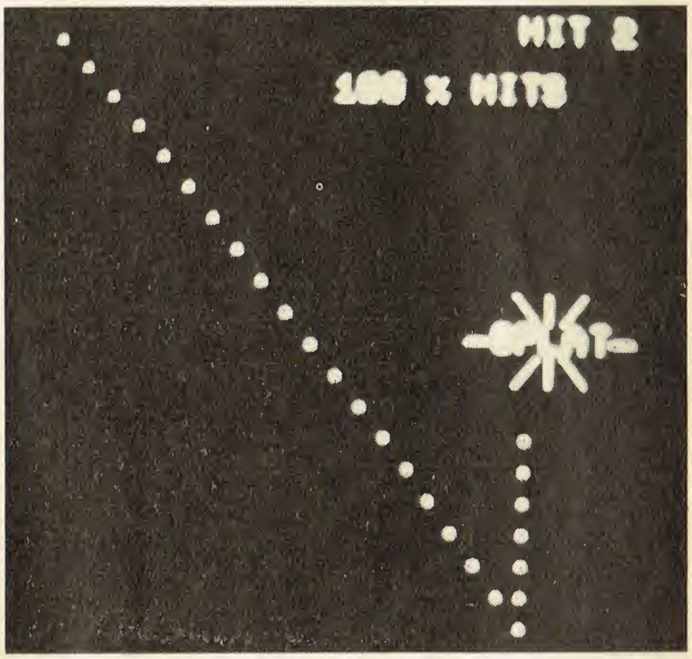
```
  1 PRINT "DO YOU WANT INSTRUCTIONS (Y
    OR N)
  2 GETA$
  3 IFA$=""THENGOTO2
  4 IFA$="Y"THENGOTO700
  5 DIMB$(8),C(8)
  6 YY=160:MM=102:SS=46:SY=32
  7 DATA"7",-41,"8",-40,"9",-39,"4",-1,"6",1,
    "1",39,"2",40,"3",41
  8 FORK=1TO8:READB$(K),C(K):NEXT
 10 REM
 15 INPUT "SET SPEED (1 TO 10) ";DF
 16 IFDF>10ORDF<1THENGOTO10
 20 DF=DF/50
100 J=32768
105 PRINT""
110 I=33267
111 POKEJ,YY
```

This is the initialisation bit i.e. printing instructions if required, setting the graphics characters for the target etc., setting up a table for "key pressed"/"direction of move" and setting the delay for the speed. J is the position of the cursor, I for the target. Line 105 is to clear the screen, but the clear screen character is not reproduced by the printer.

```
200 GETA$
210 IFA$=""THENGOTO260
212 IFA$="S"THENGOTO500
213 IFA$="N"THENGOTO900
215 POKEJ,SS
220 FORK=1TO8
230 IFA$=B$(K)THENJ=J+C(K)
235 NEXT K
240 IF J>33767THENJ=J-40
250 IF J<32768THENJ=J+40
260 POKEJ,YY
```

Moving the cursor; the direction depending on which key was pressed — the main action takes place in the loop on lines 220-235. Lines 240 and 250 stop you from going off the top and bottom of the screen.

```
265 IF RND(TI)>DF THEN 200
270 X=RND(TI)
```




```
271 POKEI-41,32
272 POKEI-40,32
273 POKEI-39,32
274 POKEI-1,32
275 POKEI,SY
276 POKEI+1,32
277 POKEI+2,32:POKEI+3,32
278 POKEI+39,32
279 POKEI+40,32
280 POKEI+41,32
281 IFX<.25THENI=I-40
290 IFX>.25ANDX<.5THENI=I—1
300 IFX>.5ANDX<.75THENI=I+1
310 IFX>.75THENI=I+40
320 IFI>33767THENI=I-40
330 IFI<32768THENI=I+40
340 POKEI,MM
341 POKEI-41,77
342 POKEI-40,66
343 POKEI-39,78
344 POKEI-1,87
345 POKEI+1,64:POKEI+2,64:POKEI+3,64
346 POKEI+39,78:POKEI+40,66
347 POKEI+41,77
350 GOTO200
```

Moving the target; the speed delay takes place at 265. Lines 271-280 blank the target, lines 281-330 move the target and lines 340-350 restore the image on the screen.

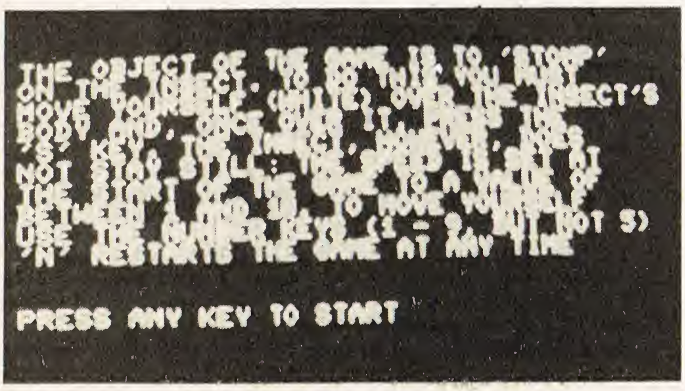
```
500 IFI=JTHENGOTO600
510 POKEJ,YY:POKEI,MM
520 PRINT"            MISSED"
```

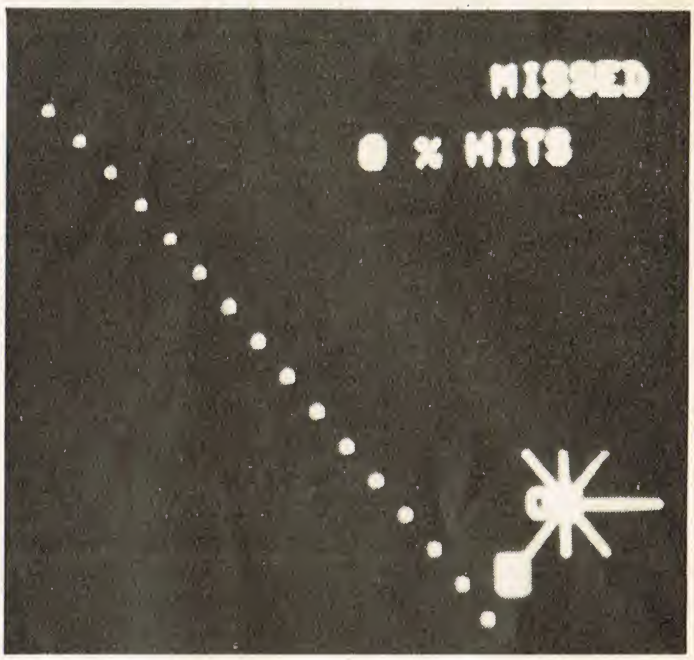
```
521 MX=MX+1
522 PRINT"   ";100*(N/(N+MX+1E-30));"%HITS"
523 FORKK=1TO1000:NEXTKK
525 PRINT""
530 GOTO270
600 N=N+1
605 POKEI-2,19:POKEI-1,16:
    POKEI,12:POKEI+1,1:POKEI+2,20
606 POKEI-3,64
610 PRINT" HIT";N
613 PRINT"        ";100*(N/(N+MX+1E-30)):
    "%HITS"
614 FORKK=1TO1000:NEXTKK
615 PRINT""
616 J=32768
620 GOTO200
```

The test for a "hit" (the position of the cursor and the position of the target coincide) is made at line 500. If they don't a message is displayed which is kept on the screen for the duration of the delay at line 523. The game then continues from where it was left off. If a "hit" has been scored then again a message is displayed, but the cursor position is reset (line 616) before continuing.

```
700 PRINT "THE OBJECT OF THE GAME IS TO
    'STOMP' "
710 PRINT "ON THE INSECT. TO DO THIS YOU
    MUST"
720 PRINT "MOVE YOURSELF (WHITE) OVER
    THE INSECT'S"
730 PRINT "BODY AND, ONCE OVER IT, PRESS
    THE"
740 PRINT " 'S' KEY. THE INSECT, HOWEVER
    DOES"
750 PRINT "NOT STAY STILL: THE SPEED IS SET
    AT"
760 PRINT "THE START OF THE GAME TO A
    VALUE OF"
770 PRINT "BETWEEN 1 AND 10. TO MOVE
    YOURSELF"
780 PRINT "USE THE NUMBER KEYS (1-9, BUT
    NOT 5)"
790 PRINT "'N' RESTARTS THE GAME AT ANY
    TIME"
800 PRINT "PRESS ANY KEY TO START"
810 GET A$
820 IFA$=""THENGOTO810
830 GOTO5
900 N=0: MX=0
950 GOTO10
```

This displays the instructions. The lines, along with lines 1-4, can be omitted if the facility is not required.





The program leaves plenty of room for improvement, and not only in its logical structure. The scoring system could easily be made more imaginative, for instance, and the intricacy of the game increased — a suggestion is to have the insect slow down if a leg is stomped off.

Geography test program

The program shown will test the geographical knowledge of the user.

The program will first prompt with a request for the area of the world that will be tested — options are Europe, South America and Asia. After this the machine will prompt with a request that the user enters the name of the capital city of one of the countries within the chosen continent. After ten questions the computer will show the score for the session and enquire whether or not the user wishes the test to continue.

The program was written for TRITON but should be suitable for implementation on most small BASIC systems.

```
 10 PRINT "GEOGRAPHY TEST—CAPITAL
    CITIES"
 20 LET R=0,W=0
 30 PRINT
 40 PRINT "1.EUROPE 2.SOUTH AMERICA
    3.ASIA"
 50 PRINT
 60 INPUT "PLEASE TYPE NUMBER OF CONTI-
    NENT CHOSEN" A
 70 IF A=1 GOTO 100
 80 IF A=2 GOTO 180
 90 IF A=3 GOTO 260
100 PRINT
110 FOR I=1 TO 12; at(I)=0; NEXT I
120 FOR L=1 TO 12
130 PRINT
140 PRINT "WHAT IS THE CAPITAL OF"; GOSUB
    340
150 GOSUB 500; INPUT "THE ANSWER IS
    NUMBER" B; GOSUB 910
160 NEXT L
```

```
170 GOTO 940
180 PRINT
190 FOR J=1 TO 10; at (J)=0; NEXT J
200 FOR M=1 TO 10
210 PRINT
220 PRINT "WHAT IS THE CAPITAL OF"; GOSUB
    550
230 GOSUB 690; INPUT "THE ANSWER IS
    NUMBER" B; GOSUB 910
240 NEXT M
250 GOTO 940
260 PRINT
270 FOR K=1 TO 10; at(K)-0; NEXT K
280 FOR N=1 TO 10
290 PRINT
300 PRINT "WHAT IS THE CAPITAL OF"; GOSUB
    730
310 GOSUB 870; INPUT "THE ANSWER IS
    NUMBER" B; GOSUB 910
320 NEXT N
330 GOTO 940
340 Z=RND(12)
350 IF at(Z)#0 GOTO 340
360 at(Z)=1
370 IF Z=1 PRINT "ITALY"
380 IF Z=2 PRINT "GREECE"
390 IF Z=3 PRINT "DENMARK"
400 IF Z=4 PRINT "E.GERMANY"
410 IF Z=5 PRINT "NORWAY"
420 IF Z=6 PRINT "PORTUGAL"
430 IF Z=7 PRINT "SPAIN"
440 IF Z=8 PRINT "SWEDEN"
450 IF Z=9 PRINT "W.GERMANY"
460 IF Z=10 PRINT "BELGIUM"
470 IF Z=11 PRINT "POLAND"
480 IF Z=12 PRINT "AUSTRIA"
490 RETURN
500 PRINT "1.ROME 2.ATHENS 3.COPENHAGEN
    4.BERLIN 5.OSLO",
510 PRINT "6.LISBON"
520 PRINT "7.MADRID 8.STOCKHOLM 9.BONN
    10.BRUSSELS 11.WARSAW",
530 PRINT "12.VIENNA"
540 RETURN
550 Z=RND(10)
560 IF at(Z)#0 GOTO 550
570 at(Z)=1
580 IF Z=1 PRINT "GUYANA"
590 IF Z=2 PRINT "PARAGUAY"
600 IF Z=3 PRINT "COLUMBIA"
610 IF Z=4 PRINT "PERU"
620 IF Z=5 PRINT "VENEZUELA"
630 IF Z=6 PRINT "ARGENTINA"
640 IF Z=7 PRINT "ECUADOR"
650 IF Z=8 PRINT "BOLIVIA"
660 IF Z=9 PRINT "URUGUAY"
670 IF Z=10 PRINT "CHILE"
680 RETURN
690 PRINT "1.GEORGETOWN 2.ASUNCION
    3.BOGOTA 4.LIMA 5.CARACAS"
700 PRINT "6.BUENOS AIRES 7.QUITO 8.LA PAZ
    9.MONTEVIDEO",
710 PRINT "10.SANTIAGO"
720 RETURN
730 Z=RND(10)
740 IF at(Z)#0 GOTO 730
750 at(Z)=1
760 IF Z=1 PRINT "MALAYSIA"
770 IF Z=2 PRINT "BURMA"
780 IF Z=3 PRINT "BANGLADESH"
790 IF Z=4 PRINT "PHILIPPINES"
800 IF Z=5 PRINT "THAILAND"
810 IF Z=6 PRINT "INDONESIA"
820 IF Z=7 PRINT "LAOS"
830 IF Z=8 PRINT "CAMBODIA"
840 IF Z=9 PRINT "IRAN"
850 IF Z=10 PRINT "IRAQ"
860 RETURN
870 PRINT "1.KUALA LUMPUR 2.RANGOON
    3.DACCA 4.MANILA 5.BANGKOK"
880 PRINT "6.JAKARTA 7.VIENTIANE 8.PHNOM
    PENH 9.TEHRAN",
890 PRINT "10.BAGHDAD"
900 RETURN
910 IF B=Z PRINT "CORRECT"; LET R=R+1
920 IF B#Z PRINT "WRONG. THE ANSWER IS
NUMBER", #3, Z; LET W=W+1
930 RETURN
940 PRINT "END OF TEST"
950 PRINT "YOUR SCORE IS",#3, R,"RIGHT
    AND",#3,W,"WRONG"
960 PRINT "OUT OF",#3,(R+W),"TURNS"
970 LET Y=0, N=1
980 INPUT "DO YOU WISH TO CONTINUE? Y OR
    N" C
990 IF C=0 GOTO 10
1000 IF C=1 PRINT "END OF PROGRAMME"
1010 STOP
```

CT

computing today marketplace

DIGITAL ALARM CLOCK MK2

Our sister magazines ETI and Hobby Electronics have sold a lot of digital alarm clocks — over 10,000 in fact — maybe that's something to do with the fact that we sell real bargains. In Computing Today we can offer you a truly modern, space age model.

It includes all the facilities expected in a good design — fast, slow setting, snooze facility, etc plus two unusual features — automatic brightness control and a weekend alarm cancel.

A version of this clock can be seen and examined at our Oxford Street Offices.

£10.50

(inclusive of VAT and Postage)

DIGITAL ALARM CLOCK MK2

TO: DIGITAL ALARM CLOCK MK2 OFFER
Computing Today
25-27 Oxford Street
London W1R 1RF

Please find enclosed my cheque/PO for £10.50 (payable to Computing Today) for my digital alarm clock.

Name ....

Address ....

....

....

Please allow 28 days for delivery.

ALARM/CHRONO LCD WATCH

Although this is our first issue of Computing Today, we have had considerable experience of marketing digital watches through our sister magazines ETI and Hobby Electronics — this is by far the most advanced and best watch we've offered to readers.

Currently it is being discounted elsewhere for typically £39.95 (we don't quote RRP as this is meaningless) and the watch is a 'chinese copy' of a very famous one in the £100 range!

The facilities are exceptional:

- Normal hours and minutes
- Continuous seconds or data display
- Day of the week
- Stopwatch with 0.1 secs resolution
- Lap time facility with automatic return to stopwatch after 6 seconds
- Different time zone setting with independent date, day of week settings
- Good bleeping alarm
- Easy time correcting: on the sixth 'pip', press a button and it's reset to 00 seconds as long as watch is plus or minus 29 seconds.

It comes with a full guarantee of course.

A sample of this watch can be seen and examined at our Oxford Street Offices.

£27.95

(inclusive of VAT and Postage)

LCD ALARM/CHRONO WATCH

TO: ALARM/CHRONO LCD WATCH OFFER
Computing Today
25-27 Oxford Street
London W1R 1RF

Please find enclosed my cheque/PO for £27.95 (payable to Computing Today) for my Alarm/Chrono LCD watch.

Name ....

Address ....

....

....

Please allow up to 28 days for delivery.

UNREPEATABLE OFFER

EPROMS
UNTESTED ERASED EPROMS
1702A (2K BITS) **£3.00**
2758 (8K BITS) (limited qty) **£10.00**

TRIACS
T2800D 8A 400V **£1.00**
T2302B 2.5A 200V **£0.60**
TIC225A 8A 100V **£0.45**

CMOS
CD4001BE (NOR GATE) **£0.14**
CD4049UBE (HEX INV BUFFER) **£0.30**
CD4050BE (HEX BUFFER) **£0.30**

TRANSISTORS
BD175 **£0.40**
2N4921 **£0.40**
BC300 **£0.25**

DIODES
1N4001 **£0.04**
1N4002 **£0.04**
1N4004 **£0.06**

MICROCOMPUTER BARGAINS

We have a stock of untested microcomputer PCB's which are surplus to our requirements. Each PCB contains an Intel 4040 (CPU), 4201 (clock), 4289 (standard memory interface), 5MHZ crystal, zero crossover detector CCT, power on reset CCT sockets for 5 x 1702A Eprom and on board power supply (50V 50 HZ input) regulated to provide 15v DC supply. These PCB's are sold as untested units with data on all chips and CCT diagram at the bargain price of **£19.00.**

Also available:—
1702A (programmed to your hexadecimal requirements) **£5.00**
4265 (general purpose input output device suitable for use with the above PCB) . . **£5.00**

Please add 28 pence P&P to your order and VAT at 8%

Cheque / Postal Order to:—

RACE ELECTRONICS LTD.
54/64, Morfa Road, Swansea
Tel. (0729) 41241/462684
Mail Order supplies only
Please allow 21 days for delivery

ETCH RESIST TRANSFER KIT SIZE 1:1

Complete kit 13 sheets 6in x 4½in **£2.50** with all symbols for direct application to P.C. board. Individual sheets 25p each. (1) Mixed Symbols (2) Lines 0.05 (3) Pads (4) Fish Plates and Connectors (5) 4 Lead and 3 Lead and Pads (6) DILS (7) BENDS 90 and 130 (8) 8—10—12 T.O.5. Cans (9) Edge Connectors 0.15 (10) Edge Connectors 0.1 (11) Lines 0.02 (12) Bends 0.02 (13) Quad in Line.

FRONT AND REAR PANEL TRANSFER SIGNS

All standard symbols and wording. Over 250 symbols, signs and words. Also available in reverse for perspex, etc. Choice of colours, red, blue, black, or white. Size of sheet 12in x 9in. Price **£1.**

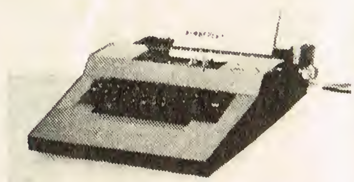
GRAPHIC TRANSFERS WITH SPACER ACCESSORIES

Available also in reverse lettering, colours red, blue, black or white. Each sheet 12in. x 9in contains capitals, lower case and numerals ⅛in kit or ¼in kit. **£1** complete. State size.

All orders dispatched promptly.
All post paid
Ex U.K. add 50p for air mail
Shop and Trade enquiries welcome
Special Transfers made to order

E. R. NICHOLLS
P.C.B. TRANSFERS
Dept. CT1
46 LOWFIELD ROAD
STOCKPORT, CHES.061-480 2179

Hard Copy



THE KIMBERLEY AZ-3 PRINTER
88 character set, ASCII coded
8cps; 11" carriage
Correspondence quality output
Price £182.86
plus £3.50 carriage + VAT

Purpose built interfaces now available for PET and TRS-80
Also text editing software

Send for full details
KIMBERLEY
Business Records
2, Hartington Road
Gosport
Hants, PO12 3AG
07017 86642

MINI-

WEST COUNTRY DEVON

CRYSTAL ELECTRONICS
CC ELECTRONICS

FOR THE BEST IN SMALL COMPUTERS
Advice and back up on Apple II, Nascom I, Atari and Newbear
BOOKS — over 150 titles
COMPONENTS, ADD ENS ETC

Apple II Stock control	**NASCOM**	
(for disc & printer) for up to 10,000 items **£100 + Vat** (full update for 6 months)	Tape containing 6 games etc **£6.00 + Vat**	All products stocked on advice from our engineers as to quality, value for money and reliability.

Shop open 0930–1800 Except Wed & Sun

40 Magdalen Rd
Torquay
Devon
Tel: 0803 22699

UNREPEATABLE offer Motorola DI microprocessor kits £32.50 incl. with milkbug, pa, ram, pcb and all components, RS232/20mA, loop, expandable, limited quantity. Also one only ready built Nascom, new £145.
C.T.S. Littleborough (0706) 79332 anytime.

EPROM PROGRAMMING prompt service 27 (1024 × 8) or 5204 (512 × 8) EPROMS programmed in your Hex (or copy) £12 each or your chips erased and programed £3 each. Special rates for batches. Pete Marlow, "Rosslyn", Moorend Grove, Cheltenham, Glos. GL53 0EX. Phone (0242) 55133 evenings.

Triton (8080)
ASKREG (listing), displays registers whilst debugging £2.
Software debugging service — send 2p per byte and commented assembly listing. Andover Software Kits, 15 Winchester Rd., Andover, Hants.

ETI Triton Computer, bought as kit, now built and fully operational. Full 4k RAM, complete with tape recorder. Offers over £375. Tel. Nottingham (0602) 266563.

2111–2 Static Ram 1 ☆£2.15 8 ★£16 16 ★£30
2112–2 Ram £2 52040 Rom £7.
Model KB756 ASCII keyboard with connector plus data £50 limited quantity, prices Inc. B. Mistry E0.8 Kenholmes Hall, Rawson Street, Leicester.

THIS SECTION IS A PRE-PAYMENT SERVICE ONLY

MINI-ADS: 31/4" x 21/8", 1-3 £38, 4-11 £36, 12 or more £34 per insertion. CLASSIFIED DISPLAY: 19p per word. Minimum 25 words. Boxed classifieds are £6.33 pec col. centimetre. No P.O. Box Numbers can be accepted without fulp address.

Enquiries to: Advertising Department, 01-437 5982. 25-27 Oxford Street, London W1R 1RF

AD INDEX

INTERESTED IN HOME COMPUTING?

FREE B BUG
valued at £23.00
plus 10 x C12 cassettes
valued at £4.00
plus Standard Modulator
valued at £2.25
WITH EVERY NASCOM

Start now and don't get left behind THE NASCOM 1 is here Ex-stock with full technical services

Plus the opportunity to join the fastest moving club of personal computer users enabling you to get the most our of your computer. You can OBTAIN and EXCHANGE programs and other software — many now available.

SAVE £60

The Powerful Z80 Microprocessor
Professional Keyboard
1 Kbyte Monitor in EPROM
2 Kbyte RAM (expandable)
Audio Cassette interface
Plugs into your domestic TV
Easy construction from straightforward instructions
— no drilling or special tools
— Just neat soldering required

Only **£197.50** + 8% VAT (includes p & p + insurance)

Manuals seperately	2.95
Z80 programming Manual	6.90
Z80 Technical Manual	2.95
PIO Technical Manual	2.95
(All prices add 8% VAT)	

NEW LOW PRICE £165.00

Power supply suitable for NASCOM 19.90

NASCOM AD ONS — Nascom improved monitor B Bug (2K) featuring — *Four times tape speed *Direct text entry without ASCII *Extended keyboard facility *Additional useful subroutines **£23.00**

Nascom Vero Case **£22.50**

Nascom Joy Stick Kit **£14.90**

Nascom Music Box Kit **£9.90**
(write your own tunes and play them on your Nascom. Complete with full documentation).

NASCOM IMMEDIATE EXPANSION S100 from COMP —strongly recommended

The only available S100 motherboard kit (fully buffered) that plugs directly into your Nascom. Designed for the insertion of S100 boards (e.g. Static RAM, EPROM and discs etc.).

S100 Motherboard/Buffer (Complete kit + documentation) £47.50
Suitable 8K Static RAM Memory ~~£125~~ £110 (fully assembled tested and guaranteed)

Buy both and get 2K Tiny Basic On cassette FREE.

NASCOM LOW COST EXPANSION

Uses dynamic RAM and NASBUS
(please note this expansion does not support S100 memory)

Tiny Basic in EPROM £25.00
8K Dynamic RAM board (in kit only) £85.00
Motherboard (in kit only) £12.50
Buffer board (in kit only) £25.00

All plus VAT

ATTENTION! TRS 80 USERS

SAVE £90

Simple to fit — only a screw-driver is required.

16K UP GRADE KIT

Half Radio Shack Price £99 + VAT

LIFETIME GUARANTEE

TRS 80 SOFTWARE *NEW*

100 MIXED PROGRAMMES on cassette **£49.00**

PET COSTS LESS AT COMP and it's a pedigree

RRP £695

The No. 1 Personal Computer in the U.K.

Affordable £590 SAVE £100

for the first time user and the professional check out the PET, the world's most popular personal computer.

THE ATARI video computer system

Atari's Video Computer System now offers more than 1300 different game variations and options in twenty great Game Program^{TM} cartridges!

Have fun while you sharpen your mental and physical coordination. You can play rousing, challenging, sophisticated video games, the games that made Atari famous.

You'll have thrill after thrill, whether you're in the thick of a dogfight, screeching around a racetrack, or dodging asteroids in an alien galaxy. With crisp bright color (on color TV) and incredible, true-to-life sound effects. With special circuits to protect your TV.

Cartridges now available
Basic Maths, Airsea Battle, Black Jack, Breakout, Surround, Spacewar, Video Olympics, Outlaw, Basketball, All at £13.90 each.

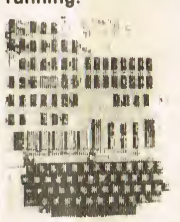
Years and years of fun and satisfaction are assured

SAVE £30

~~£169~~ **£139.00**

OHIO SUPERBOARD II *NEW*

For electronic buffs. Fully assembled and tested. Requires +5V at 3 Amps and a video monitor or TV with RF converter to be up and running.



STANDARD FEATURES
Uses the ultra powerful 6502 microprocessor
8K Microsoft BASIC-in-ROM
Full feature BASIC runs faster than currently available personal computers and all 8080-based business computers.
4K static RAM on board expandable to 8K
Full 53-keyboard with upper/lower case and user programmability
Kansas City standard audio cassette interface for high reliability
Full machine code monitor and I/O utilities in ROM

Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters; without overscan up to 30 x 30 characters.

EXTRAS
Available expander board features 24K static RAM (additional), dual mini-floppy interface, port adapter for printer and modem and an OSI 48 line expansion interface.
Assembler/editor and extended machine code monitor available.

~~£289.00~~ **£249.00** + VAT

Send £10 to reserve one - pay balance on delivery.

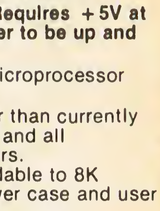
MODULATORS UHF Channel 36

Standard 6 meg band width **£2.25**
High Quality 8 meg band width £4.90

BULK PURCHASE VDU MONITORS

~~£99~~ **£69** + VAT

CASED AND GUARANTEED *12"*



SPRING SUPER SALES OFFERS

NL NewBear

Computing Store

SPECIAL OFFER!
Buy a Shugart disc drive, get the controller chip (INS 1771) for only £30.

Keyboards

* Star Devices capacative touch keyboard £37.50 inc VAT + p&p*
63 Key Ascii Encoded Kit £56.00
63 Key 'ASR 33' Style Hitex Keyboard unencoded £27.50
U.V. Prom Eraser £56.00
77-68 The best supported hobbyist 6800 system in the U.K. available as complete kits or as individual components.

Bearbag 1	77-68 CPU PCB and components	£ 45.00
Bearbag 2	77-68 LEDS and Switches	£ 14.95
Bearbag 3	77-68 Power Supply	£ 17.95
Bearbag 4	77-68 19" Rack and Backplane	£ 27.70*
Bearbag 5	77-68 4K Ram PCB and Components	£ 75.00
Bearbag 6	77-68 Mon 1 PCB and Components	£ 50.70
Bearbag 12	77-68 VDU PCB and Components	£ 69.50
Bearbag 13	77-68 Mon 2 PCB and Components	£ 64.10
Bearbag 16	77-68 Rom A (Eprom Board) and Components	£ 29.50
Bearbag 17	77-68 P10 board and Components	£ 45.00

Other Bearbags Available

Bearbag 7	4K Ram PCB and components Exorciser compatible	£ 71.50
Bearbag 8	8K Ram PCB and components Exorciser compatible	£160.00
Bearbag 9	Petiterid VDU Kit	£ 85.00
Bearbag 10	Kansas City Cassette Interface	£ 18.95
Bearbag 11	UHF Modulator	£ 3.50
Bearbag 14	2708 Prom Programmer (6800)	£ 35.00
Bearbag 18	Cottis Blandford Cassette Interface	£ 17.25

Horizon

(S100 bus Z80 based micro)

16K RAM with Single Disc Drive	£1250
32K RAM with Double Disc Drive with 2 Serial and 1 parallel port	£1975

SYM-1

The 6502 based Micro from Synertek
* 4K monitor
* Up to 4K RAM on board
* High speed cassette interface
* Fully assembled and tested £199.00 + 8% VAT

8K Basic for SYM-1 in ROM £147.00

Sorceror from Exidy

8K RAM with cassette interface £650.00
Extensive range of 6800 Software on cassette, Hardware components and Books — Send or phone for up-to-the-minute information!
Quantity discounts available on memories!

Floppy Disc Drives

8" DRI Model 7100 Single Sided Disc Drive	£325.00 carriage £5
8" DRI Model 7200 Double Sided Disc Drive	£365.00 carriage £5.00
5¼" Shugart SA400 Disc Drive	£225.00 carriage £4.50

S100 Microbox £195.00

(6 Slot card cage with power supply and motherboard with 3 sockets fitted)

ITHACA AUDIO S100 BOARDS NOW IN STOCK!

SEND FOR AN UP TO DATE CATALOGUE

Postage and packing 50p
*Postage and packing £1.50

Minimum official order £10.00. Send for catalogue to Newbear Computing Store, Bone Lane, Newbury, Berks. Callers welcome Mon-Sat 9.00-5.30 but please phone us first on 0635 49223. Visit our Northern office at 2A Gatley Road, Cheadle Cheshire. Tel 061-491-2290.
ALL MAIL ORDER TO NEWBURY.
Barclaycard and Access Welcome.

24 HOUR DELIVERY IN MOST CASES